Array List

Task 2: Interview Questions

Java ArrayList class uses a dynamic array for storing the elements. It is like an array, but there is no size limit. We can add or remove elements anytime. So, it is much more flexible than the traditional array. It is found in the java.util package. It is like the Vector in C++.

The ArrayList in Java can have the duplicate elements also. It implements the List interface so we can use all the methods of the List interface here. The ArrayList maintains the insertion order internally.

It inherits the AbstractList class and implements List interface.

Illustration

ArrayList Integer Object Type :

2	5	12	1	79	11
0	1	2	3	4	5

Integer type (for all indices) Data

Features

- Java ArrayList class can contain duplicate elements.
- Java ArrayList class maintains insertion order.
- Java ArrayList class is non synchronized.
- Java ArrayList allows random access because the array works on an index basis.
- In ArrayList, manipulation is a little bit slower than the LinkedList in Java because a lot of shifting needs to occur if any element is removed from the array list.
- We cannot create an array list of the primitive types, such as int, float, char, etc. It is required to use the required wrapper class in such cases. For example:

    ArrayList<int> al = ArrayList<int>(); // does not work

    ArrayList<Integer> al = new ArrayList<Integer>(); // works fine

- Java ArrayList gets initialized by the size. The size is dynamic in the array list, which varies according to the elements getting added or removed from the list.

ArrayList Example

Example 1:

```
import java.util.*;
 public class ArrayListExample1{
 public static void main(String args[]){
```

```java
    ArrayList<String> list=new ArrayList<String>();  //Creating arraylist
       list.add("Mango");  //Adding object in arraylist
       list.add("Apple");
       list.add("Banana");
       list.add("Grapes");
       //Printing the arraylist object
       System.out.println(list);
   }
}
```

Output

[Mango, Apple, Banana, Grapes]

Example 2:

```java
import java.util.*;
public class ArrayListExample2{
 public static void main(String args[]){
  ArrayList<String> list=new ArrayList<String>(); //Creating arraylist
  list.add("Mango"); //Adding object in arraylist
  list.add("Apple");
  list.add("Banana");
  list.add("Grapes");
  //Traversing list through Iterator
  Iterator itr=list.iterator();//getting the Iterator
  while(itr.hasNext()){//check if iterator has the elements
   System.out.println(itr.next());//printing the element and move to next
  }
 }
}
```

Output:

Mango
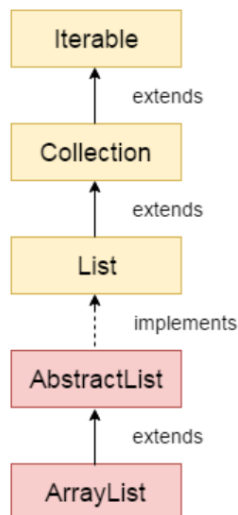
Apple

Banana

Grapes

Interview Questions

1. Explain the hierarchy of ArrayList class in Java.

As shown in the above diagram, the Java ArrayList class extends AbstractList class which implements the List interface. The List interface extends Collection and Iterable interfaces in hierarchical order.

2. Which collection classes implement List interface in Java?

The collection classes that implement List interface, are as:

- ArrayList
- LinkedList
- CopyOnWriteArrayList
- Vector
- Stack

3. What are the important features of ArrayList in Java?

There are several significant features of ArrayList in Java that are as follows:

- ArrayList in Java uses an index-based structure.
- The size of ArrayList can increase or decrease at runtime. Once ArrayList is created, we can add any number of elements.
- An ArrayList allows adding elements into the middle of collection.
- It allows to delete elements.
- Duplicate elements are allowed in the array list.
- Any number of null elements can be added to ArrayList.

- ArrayList maintains the insertion order in Java. That is insertion order is preserved.
- ArrayList is not synchronized. That means multiple threads can use the same ArrayList objects simultaneously.
- Since ArrayList implements random access interface, we can get, set, insert, and remove elements of the array list from any arbitrary position.
- The performance of ArrayList is slow because if any element is removed from ArrayList, a lot of shifting takes place.

4. Why is ArrayList called a dynamically growing array in Java?

ArrayList is called a dynamically growing array in java because ArrayList uses a dynamic array internally for storing a group of elements.

If the initial capacity of the array is exceeded, a new array with a larger capacity is created automatically and all the elements from the current array are copied to the new array.

When elements are removed from the array list, the size of array list can be shrunk automatically.

5. What is the difference between the length of an array and size of ArrayList?

The length of an array can be determined by using property length. But ArrayList does not support the length property. It provides size() method that can be used to find the number of elements in the list.