

Name : Tushar Popat Kakad

Batch : PPA10

Date : 8/12/2022

```
import java.util.ArrayList;
```

```
class Test
```

```
{
    public static void main(String[] args)
    {

        // COLLECTION METHODS

        // 1. add() method
        ArrayList al = new ArrayList();
        al.add("Tushar") ;
        al.add("Kakad");
        System.out.println(al);

        ArrayList al2 = new ArrayList();
        al2.add("Sangamner") ;
        al2.add("Nagar");
        System.out.println(al2);

        // 2. contains method
        System.out.println("Contains method : "+ al.contains(al2));

        // 3. containsAll method
        System.out.println("Contains all method : "+ al.containsAll(al2));

        // 4. addAll() method
        al.addAll(al2);
        System.out.println("addAll method : "+al);

        // 5. remove() method
        al.remove("kakad");
        System.out.println("remove method : " +al);

        // 6. removeAll() method
        al.removeAll(al2);
        System.out.println("removeAll method: "+al);

        // 7. size() method
        al.size();
        System.out.println("The size of an array is : " + al.size());

        // 8. isEmpty() method
        al.isEmpty();
        System.out.println("isEmpty method : " + al.isEmpty());
    }
}
```

```
// 9. clear() method
al.clear();
System.out.println("Clear method : "+al);

// 10. retainAll() method
al.retainAll(al2);
System.out.println("Retain all method : " + al.retainAll(al2));

// 11. toArray() method
Object[] obj = al.toArray();
System.out.println(al.toArray());
```

// LIST INTERFACE METHODS

```
// 1. add() method
ArrayList al3 = new ArrayList();
al3.add("Virat");
al3.add("Kohli");
System.out.println(al3);

ArrayList al4 = new ArrayList();
al4.add("Rohit");
al4.add("Sharma");
System.out.println(al4);

// 2. addAll() method
al3.addAll(al4);
System.out.println(al3);

// 3. set() method
Object obj2 = al3.set(1,"Dhoni");
System.out.println(obj2+" is replaced");
System.out.println(al3);

// 4. get() method
Object obj3 = al3.get(3);
System.out.println(obj3);

// 5. indexOf() method
int i = al3.indexOf("Dhoni");
System.out.println(i);

// 6. lastIndexOf() method
int i2 = al3.lastIndexOf("Dhoni");
System.out.println(i2);

// 7. remove() method
Object obj4 = al3.remove(2);
System.out.println(obj4+" is removed");
System.out.println(al3);
```

```
// ARRAY LIST CONSTRUCTORS
```

```
// Default constructor
```

```
ArrayList al5 = new ArrayList();
```

```
// Constructor with capacity
```

```
ArrayList al6 = new ArrayList(10);
```

```
//Constructor with collection
```

```
ArrayList al7 = new ArrayList(al5);
```

```
    }  
}
```


SR NO collection (1-2)

MAP. (1-2)

1) collection can store data without identity

MAP can store data with identity.

2) collection uses Array Format to store data

MAP uses key-value pair format to store data.

3) before collection there was vector framework in Java (1.0)

before collection there was Dictionary framework in Java (1.0)

4) vector is synchronized and it executes in

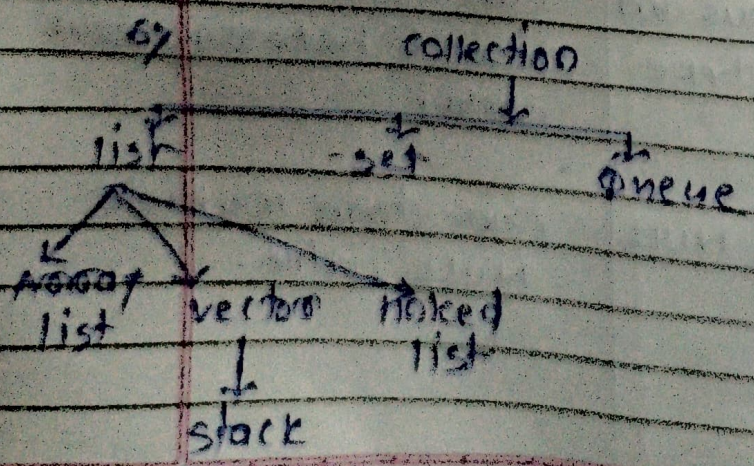
MAP is also synchronized & its execute in sequential.

sequential. it's a drawback of vector.

It is a drawback of Dictionary

5) To overcome the drawback & store element in Array or index based format we use collection

To overcome the drawback & store element in key-value pair format we use MAP.



Sr No	Array list	Vector
1	It is not a legacy collection	It is a legacy collection
2	It has direct implementation from list	It has direct implementation from list.
3	Indirect implementation from collection	Indirect implementation implementation from collection.
4	Data stored in index-based format	Data stored in index-based format.
5	Can store duplicate elements	Can store duplicate element
6	It preserves the insertion order.	It also preserves the insertion order.
7	Sorting order doesn't preserve	Sorting order doesn't preserve.
8	Can store homogeneous and heterogeneous both type of data.	Can store both homogeneous & heterogeneous data.
9	Can store multiple NULL value.	Can store multiple NULL value.

SR No	Array list	vector
10	It supports resizable array	It supports resizable array
11	Formula = $(\text{current_capacity} * \frac{3}{2}) + 1$	Formula = $(\text{current_capacity} * 2)$
12	We can use this for frequent retrieval of data	We can use this for frequent retrieval of data.
13	It is Non-synchronized Interface.	It is Synchronized Interface
13 a)	All methods are Asynchronous	All methods are Synchronized
13 b)	Execution is concurrent	Execution is sequential
13 c)	Reduce execution time	Increases execution time
13 d)	Increases performance	Reduces performance
13 e)	It may occur data inconsistency	It can provide data consistency
13 f)	No Thread-Safe	Thread-Safe.

No	List	Set
17	Data stored in index based format.	→ Data stored in hashcode based format.
27	It allows to store duplicate element	It doesn't allow to store duplicate element
37	list preserves insertion order	set doesn't preserve insertion order
47	list doesn't preserve sorted order	→ set doesn't preserve sorted order
57	can store multiple NULL values	Can't store multiple NULL values.
67	List can store both Homogeneous & Heterogeneous type of data	set can store only Homogeneous type of data.