List Interface:
Extends the collection interface and represents an ordered collection of elements.

- Allows the duplicate elements
- Maintains the insertion order
- Allows multiple null values

- List Implementations:
  1. Arraylist
  2. LinkedList

3. Vector

ArrayList – [ArrayList (Java SE 21 & JDK 21)]

Characteristics:

1. Uses dynamic array internally
2. It provides faster random access using indices
3. Not synchronized, it’ s not thread safe
4. To make it threadsafe [Collections (Java SE 21 & JDK 21)]
   In collections utility class ->
   Collections.synchronizedList({name of list});

```
package arrayListExample;

import java.util.ArrayList;

public class ArrayListEx {
    public static void main(String[] args) {
        ArrayList<String> names = new ArrayList<>();
        //Adding an element in a ArrayList
        names.add("Krishna");
        names.add("Gopal");
        names.add("Govind");
        names.add("Parth");

        //Accessing the first element
        System.out.println("First Element: "+ names.get(0));

        //Iterating elements
        for (int i=0; i < names.size(); i++){
            System.out.println(names.get(i));
        }

        //Removing an element
        names.remove("Parth");

        //After removal printing the list
        System.out.println("After removal of an element: "+
names);
    }
}
```

Methods:

Add(): adds an element to the list

Get(int index): retrieves the element at that index

Remove()

Size()

Contains(Object o): to check if a list contains an element or not.

Clear()

`set(int index, E element)`: Replaces the element at the specified position in this list with the specified element.

LinkedList - [LinkedList (Java SE 21 & JDK 21)]

Characteristics:

- Internally uses doubly linked list structure
- Efficient for insertions and deletions

```java
package LinkedExample;

import java.util.LinkedList;

public class LinkedListExample {
    public static void main(String[] args) {
        LinkedList<String> tasks = new LinkedList<>();

        //adding elements
        tasks.add("READ");
        tasks.add("WRITE");
        tasks.add("Exercise");

        //Adding elements at specified elements
        tasks.addFirst("Wake Up");
        tasks.addLast("Sleep");

        System.out.println("After Adding elements at first
and last: "+ tasks);

        //to access the first task
        System.out.println("First Task: "+
tasks.getFirst());

        //to remove first and last elements
        tasks.removeFirst();
        tasks.removeLast();

        //After removal
        System.out.println("After Removal: "+ tasks);
    }
}
```

Methods:

addFirst(): adds the element at the beginning

addLast(): adds the element at last

getFirst()/getLast(): retrieves the first or last element

add(int index, E element): Inserts the element at specified position

Vector – Vector (Java SE 21 & JDK 21)

Characteristics:

Similar to arraylist, but it is synchronized(thread safe)

Slower performance as compare to arrayList due to synchronization.

It is automatically resizable.

```java
package vectorEx;

import java.util.Vector;

public class VectorExample {
    public static void main(String[] args) {
        Vector<Integer> numbers = new Vector<>();

        //Adding Elements
        numbers.add(10);
        numbers.add(20);
        numbers.add(30);
        numbers.add(40);
        numbers.add(50);
        numbers.add(60);
        numbers.add(70);
        numbers.add(80);
        numbers.add(90);
        numbers.add(100);
        numbers.add(110);
        numbers.add(120);
        numbers.add(130);
        numbers.add(140);
        numbers.add(150);
        numbers.add(160);
        numbers.add(170);
        numbers.add(180);
        numbers.add(190);
        numbers.add(200);
        numbers.add(210);

        //to access the element
        System.out.println("First Element: "+ numbers.get(0));

        System.out.println("Vector Elements: "+ numbers);

        System.out.println("Vector Capacity: "+
numbers.capacity());
```

```
        }
}
```

https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/System.html - System Class