**04-05-202.**

**Collection:-**

* It is a framework that provide an architecture to store and manipulate the group of objects.
* Manipulation means delection,sorting,updation,searching etc.
* It is interface which is extend the iterable interface.

collections also as sub interfaces those are

1. **List.**
2. **Queue.**
3. **Set**.

**06-05-2022.**

**Hierarchy of Collection Framework**



**09-05-2022.**

**List:-**

* List is an Interface which extends **Collection Interface**.
* List allows duplicate values dynamically.
* List has an order.
* We can send bulk amount data into list.

**List <Data Type> Reference Name = new ArrayList();**

**List <Data Type> Reference Name = new LinkedList();**

**List <Data Type> Reference Name = new Vector();**

**List <Data Type> Reference Name = new Stack();**

List interface is implemented in three class those are

1. **ArrayList(class).**
2. **LinkedList(class).**
3. **Vector(class).**

**Iterable:**

It is an interface, but this interface is the roof of the all collection classes.

Iterable extends by the collection.

Iterable will be implemented by all the sub classes.

Iterable is having only one method.

**Iterator <T> = iterator( );**

**Array List:-**

* It is a dynamic array which store the data and has duplicate elements od the different data types.
* This class will maintain an order and it is non synchronized.
* We can access the elements stored in the array list randomly.
* The deletion is to slow.

**List <Data Type> Reference Name = new ArrayList();**

(or)

**ArrayList <Data Type> Reference Name = new ArrayList();**

**10-05-2022.**

**LinkedList:-**

* It maintains insertion order.
* It allows duplicates and null values.
* It is not synchronized.
* Manipulation is fast.
* It can be used as **List, Stack, Queue**.
* It uses doubly linked list to store elements.

**Hierarchy for LinkedList Class:List (I) Queue(I) | | | Implements | Extends | |AstractSequentialList (C) Deque(I) | | | Extends | Implements | |LinkedList (C) LinkedList (C)**

|  |  |
| --- | --- |
| **ArrayList** | **LinkedList** |
| 1.ArrayList Internally uses dynamically array To store the elements.so that array is resizable Or growable. | 1.LinkedList internally uses doubly LinkedList to store the elements. |
| 2.ArrayList is for searching/ accessing the data | 2.It is best insertion or deletion. |
| 3.Insertion or deletion with Arraylist is slow because it internally use array. if any element is removed from the array, all the bits are shifted In memory. | 3.Insertion or deletion with linkedlist is faster than arraylist because no bit shifting is required in memory. |
| 4.ArrayList takes less memory overhead as it stores only objects | 4.LinkedList takes more memory overhead ,as it stores the object as well as the address of the object. |
| 5.ArrayList class can acts as a list only because it implements list only. | 5.LinkedList class can act as a list and queue both because it implements list and deque interface. |
| 6.ArrayList Provides random access. | 6.LinkedList does not provide random access. |