Q.Explain collection hierarchy? Collection framework hierarchy?

Java.util collection is the root of the java collection and most of the collection is inheriteqd from this interface. So the Collection is the parent interface which has the 3 child interfaces are List interfaces, Set Interfaces Qeue interface

List interaces has implementation class by few classes arraylist, linked list ,vector and vector class extends by stack class

Set interfaces extended by two implementation class hashset and linkedhashset again set is extended by the sorted set which is interface which have the implementing class tree set

Queue interface has only one implementing class that is prority queue and again Queue extended by the Deque interface which has the implementing class arraydeque

Q.What is collection?

Java collection is framework that provides an architecture to store and manipulate group of objects It is single unit of object

Q.Why the map doesn’t extends collection framework?

Map interface in java follows key value pair structure whereas the collection interface is a collection of object which stored in structured manner with a specified access mechanism

The main reason map doesn’t extends collection framework is that add(E,e) method of collection framework does not support key- value pair like map interface put(K, V) method

It might not extend collection framework but still it an integral part of the collection framework

Q.What is Collection framework

The Collection framework represents a unified architecture for storing and manipulating a group of objects. It has: Collection Framework is a combination of classes and interface, which is used to store and manipulate the data in the form of objects. It provides various classes such as ArrayList, Vector, Stack, and HashSet, etc. and interfaces such as List, Queue, Set, etc. for this purpose.

1. Interfaces and its implementations, i.e., classes
2. Algorithm

Q. Difference between collection and array?

1)Array is nothing but group of multiple element of similar data type and Collection is nothis but the single unit of object

2) array has fixed size whereas collection are dynamic in size

3)It stores primitive datat type and wrapper class object whereas collection stores only object

4)Array don’t have ready made method wheareas Collection have a ready made data structure AND methods

5)In case of mamemory management array not good whereas collection are good in case of memory management

Q.**Explain various interfaces used in Collection framework?**

Collection framework implements various interfaces, Collection interface and Map interface (java.util.Map) are the mainly used interfaces of Java Collection Framework. List of interfaces of Collection Framework is given below:

1. **Collection interface:** Collection (java.util.Collection) is the primary interface, and every collection must implement this interface.
2. **2. List interface:**List interface extends the Collection interface, and it is an ordered collection of objects. It contains duplicate elements. It also allows random access of elements.
3. **Set interface:** Set (java.util.Set) interface is a collection which cannot contain duplicate elements. It can only include inherited methods of Collection interface

4.**Queue interface:**Queue (java.util.Queue) interface defines queue data structure, which stores the elements in the form FIFO (first in first out).

5 **. Dequeue interface:** it is a double-ended-queue. It allows the insertion and removal of elements from both ends. It implants the properties of both Stack and queue so it can perform LIFO (Last in first out) stack and FIFO (first in first out) queue, operations.

**6. Map interface:**A Map (java.util.Map) represents a key, value pair storage of elements. Map interface does not implement the Collection interface. It can only contain a unique key but can have duplicate elements. There are two interfaces which implement Map in java that are Map interface and Sorted Map.

List Interface

Q.What is list interface in java? List interface is the child interface of Collection interface. It inhibits a list type data structure in which we can store the ordered collection of objects. It can have duplicate values. It maintain insertion order

List interface is implemented by the classes ArrayList, LinkedList, Vector, and Stack.

.

Q.What is arrayList?

The ArrayList class implements the List interface. It uses a dynamic array to store the duplicate element of different data types. The ArrayList class maintains the insertion order and is non-synchronized. The elements stored in the ArrayList class can be randomly accessed

Java **ArrayList** class uses a dynamic [*array*](https://www.javatpoint.com/array-in-java) 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](https://www.javatpoint.com/java-list).

* Java ArrayList class can contain duplicate elements.
* Java ArrayList class maintains insertion order.
* Java ArrayList class is non [synchronized](https://www.javatpoint.com/synchronization-in-java).
* 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 can not 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:

1. ArrayList<**int**> al = ArrayList<**int**>(); // does not work
2. 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.

QDifference between linked list and array list?

|  |  |
| --- | --- |
| **ArrayList** | **LinkedList** |
| 1) ArrayList internally uses a **dynamic array** to store the elements. | LinkedList internally uses a **doubly linked list** to store the elements. |
| 2) Manipulation with ArrayList is **slow** because it internally uses an array. If any element is removed from the array, all the other elements are shifted in memory. | Manipulation with LinkedList is **faster** than ArrayList because it uses a doubly linked list, so no bit shifting is required in memory. |
| 3) An ArrayList class can **act as a list** only because it implements List only. | LinkedList class can **act as a list and queue** both because it implements List and Deque interfaces. |
| 4) ArrayList is **better for storing and accessing** data. | LinkedList is **better for manipulating** data. |
| 5) The memory location for the elements of an ArrayList is contiguous. | The location for the elements of a linked list is not contagious. |
| 6) Generally, when an ArrayList is initialized, a default capacity of 10 is assigned to the ArrayList. | There is no case of default capacity in a LinkedList. In LinkedList, an empty list is created when a LinkedList is initialized. |
| 7) To be precise, an ArrayList is a resizable array. | LinkedList implements the doubly linked list of the list interface. |

Q.What is linkedlist?

A [linked list](https://www.geeksforgeeks.org/linked-list-set-1-introduction/) is a linear data structure that contains nodes and pointers. Each node contains data and pointers to another node. It is an ordered collection of data elements called nodes and the linear order is maintained by pointers. Like an array, a linked list also contains elements of homogeneous data types.

Java LinkedList class uses a doubly linked list to store the elements. It provides a linked-list data structure. It inherits the AbstractList class and implements List and Deque interfaces.

The important points about Java LinkedList are:

* Java LinkedList class can contain duplicate elements.
* Java LinkedList class maintains insertion order.
* Java LinkedList class is non synchronized.
* In Java LinkedList class, manipulation is fast because no shifting needs to occur.
* Java LinkedList class can be used as a list, stack or queue.

Q.When we have to use arrayList?

ArrayList provides constant time for search operation, so it is better to use ArrayList if searching is more frequent operation than add and remove operation.

* Arrays are used when we require random access to elements.
* Arrays are used as the base of all sorting algorithms.
* Arrays are used when the number of elements(size of an array) is known in advance as the array supports static memory allocation.
* Arrays are used to implement matrices.
* Arrays can be used to implement various data structures like stacks, queues, and tree

Q.When we have to use LinkedList?

The LinkedList provides constant time for add and remove operations. So it is better to use LinkedList for manipulation.

* Linked Lists are used when the number of elements is not known in advance i.e. size is not known as linked lists support dynamic memory allocation.
* Linked lists are simple and can be used to implement other data structures like stack, queue, and tree.
* Linked Lists can be used for the manipulation of polynomials.
* Linked lists are used for performing arithmetic operations on long integers.
* Linked List can be used in cases when faster insertion and deletion are required. Linked takes O(1) time complexity for insertion and deletion while array takes O(N).

Q.Difference between Set and List?

|  |  |  |
| --- | --- | --- |
| **.No** | **List** | **Set** |
| 1. | The list implementation allows us to add the same or duplicate elements. | The set implementation doesn't allow us to add the same or duplicate elements. |
| 2. | The insertion order is maintained by the List. | It doesn't maintain the insertion order of elements. |
| 3. | List allows us to add any number of null values. | Set allows us to add at least one null value in it. |
| 4. | The List implementation classes are LinkedList and ArrayList. | The Set implementation classes are TreeSet, HashSet and LinkedHashSet. |
| 5. | We can get the element of a specified index from the list using the get() method. | We cannot find the element from the Set based on the index because it doesn't provide any get method(). |
| 6. | It is used when we want to frequently access the elements by using the index. | It is used when we want to design a collection of distinct elements. |
| 7. | The method of List interface listiterator() is used to iterate the List elements. | The iterator is used when we need to iterate the Set elements. |

Q.What is data structure behind linkedlist?

A linked list is **a linear data structure.** **It is a collection of nodes, and a node contains data and addresses the next node**. Linked lists do not use contiguous memory allocation for storage,

Q.What is data structure behind ArrayList?

ArrayList used dynamic data structure to store element internally and shiffiting require too much if element is remove fro an array random access will be done based on index manner