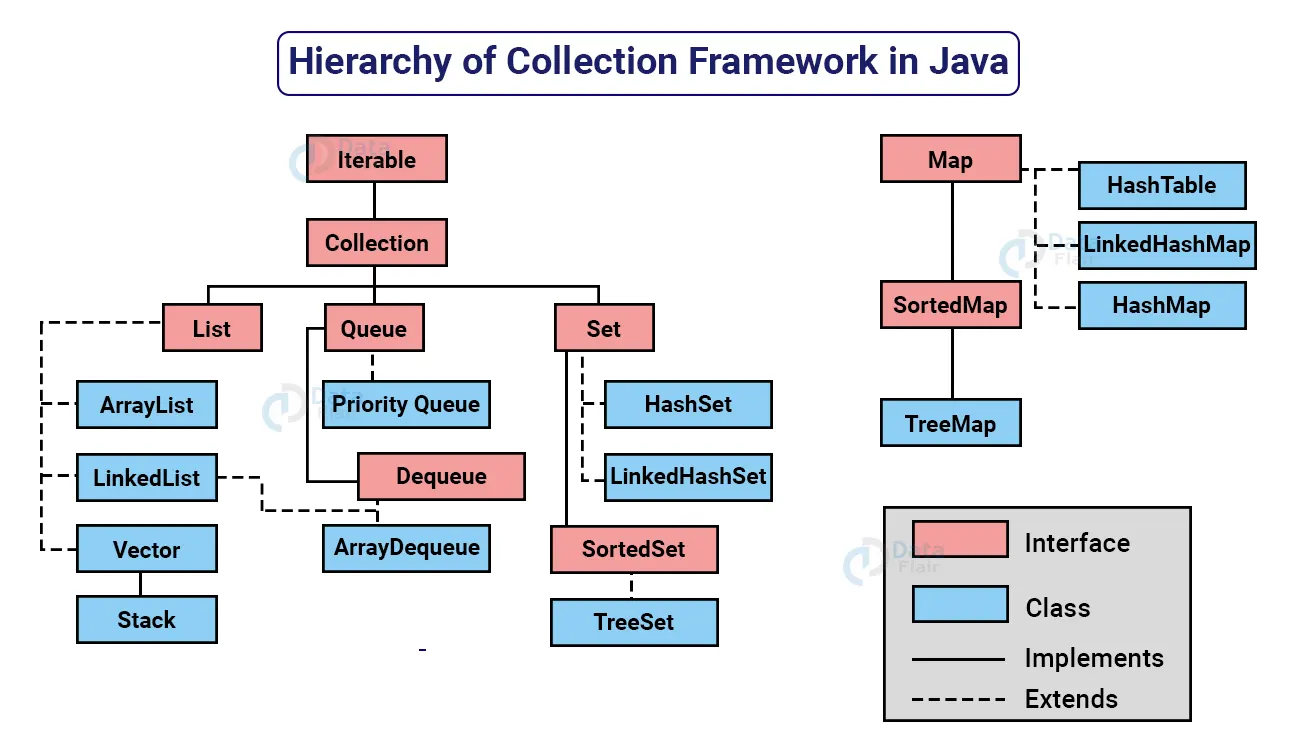
Collection:



what is collection framework ?

The Java collections framework **provides a set of interfaces and classes to implement various data structures and algorithms**.

iterable interface in java:

The **Iterable**interface was introduced in JDK 1.5. It belongs to **java.lang**package. In general, an object Implementing Iterable allows it to be iterated. An iterable interface allows an object to be the target of [enhanced for loop](https://www.geeksforgeeks.org/for-each-loop-in-java/)(for-each loop).

what is collection Interface?

The **Collection**interface is a member of the [Java Collections Framework](https://www.geeksforgeeks.org/collections-in-java-2/). It is a part of **java.util** package. It is one of the root interfaces of the Collection Hierarchy. The Collection interface is not directly implemented by any class. However, it is implemented indirectly via its subtypes or subinterfaces like [List](https://www.geeksforgeeks.org/list-interface-java-examples/), [Queue](https://www.geeksforgeeks.org/queue-interface-java/), and [Set](https://www.geeksforgeeks.org/set-in-java/).

what is queue in java?

A queue is **an object that represents a data structure designed to have the element inserted at the end of the queue, and the element removed from the beginning of the queue**. Java. Util. Queue contains multiple elements before the process. The order of elements of the queue in Java is FIFO (first-in-first-out).

what is stack in java ?

The stack is **a linear data structure that is used to store the collection of objects**. It is based on Last-In-First-Out (LIFO). Java collection framework provides many interfaces and classes to store the collection of objects.

what is vector in java?

**Vector is similar to ArrayList but the differences are**, it is synchronized and its default initial size is 10 and when the size exceeds its size increases to double of the original size that means the new size will be 20. Vector is the only class other than ArrayList to implement RandomAccess.

what is hashtable in java ?

It is similar to HashMap, but is synchronized. **Hashtable stores key/value pair in hash table**.

what is hashmap in java ?  
The HashMap class of the Java collections framework **provides the functionality of the hash table data structure**. It stores elements in key/value pairs. Here, keys are unique identifiers used to associate each value on a map. The HashMap class implements the Map interface.

list:  
 1)duplicates are allowed  
  2)insertion order is maintained  
 package;java.util.ArrayList

ArraylList

1. data retrievel is  faster  
   2.data insertion and delte is slower  
   ArraylList ar=new ArrayList();  
   ar.add(10);  
   ar.add(1);  
   ar.contains(1);------ans : true\  
   ArraylList ar2=new ArrayList();  
   ar2.addAll(ar); move value from one array to another array  
   ar2.add(2);--- ans : 10,1,2 ----length=3;  
     
   {add 14 ,index 1;  
    ar2.add(index,value) }  
   ar2.add(1,14);---------10,14,1,2  
   ar2.size();--- total lenhth=4

{  get value  
  ex;ar2.get(index) }

ar2.get(2);----ans:1;  
Object o=ar2.get(2);  
 o = ----ans:1;

{remove value;  
ex;ar2.remove(index);}  
   
ar2.remove(0);---14,1,2  
   
ar2.removeAll();

{ find value index num  
 ex;ar2.indexOf(value);-----3,7,1,4,9,7 }  
  
ar2.indexOf(4);--ans:3  
ar2.indexOf(7);--ans:1   
ar2.lastIndexOf(7);--ans:5  
  
data retrievel is faster  
data insertion and delte is slower  
  
 LinkedList:  
  
LinkedList ar2 =new LinkedList();  
  
1,data retrievel is  slower  
2,data insertion and delte is faster

LinkedList ar2 =new LinkedList();  
  
   Arrays  
   Arrays.sort(arr name)

  SET--->

1. duplicates are not allowed  
   2)insertion order is not maintained  
   Hash set  
   1)duplicates are not allowed  
   2)insertion order is not maintained

LinkedHashSet   
1)duplicates are not allowed

1. insertion order is  maintained

set---sortset--treeset

tree set------ isertion order not maintain, homogeneous element,no duplicates,  
  //print value is assending order  
================================================ 

Map :

MAP----COMBININATION OF KEY AND VALUE  
KEY-- NO DUPLICATES  
VALUE---DUPLICATES  
KEY IS OBJECT  
VALUE IS OBJECT  
combination of each key and value entry--key value pair  
tamil=22  
siva=21

vimal=22

collection of entries map  
 ===========================================  
HashMap hm=new HashMap();  
hm.put(poori,40);  
 hm.put(pogal,78);  
 output-----{poori=20,pogal=78}  
  =====================================================  
  LinkedHashMap hm=new LinkedHashMap();  
  hm.put(poori,40);  
  hm.put(pogal,78);  
  output-----{poori=20,pogal=78}  
  ===============================================  
  TreeMap hm=new TreeMap();  
  hm.put(poori,40);  
  hm.put(pogal,78);  
  output-----{poori=20,pogal=78}  
 ==============================================  
  HashMap----insertion order not maintain  
  LinkedHashMap----insertion order  maintain  
  TreeMap---key is ascending order because key no duplicates  
  =========================================  
  HashMap hm=new HashMap();  
    hm.put(poori,40);  
    hm.put(pogal,78);  
    output-----{poori=40,pogal=78}  
    Entry is a inner interface  //METHOD  
  hm.put(key,value);  
  hm.containsKey("poori");----true  
  hm.containsvalue(20);----false  
  hm.keySet();-----poori,pogal  
  hm.values();---------20,78  
  hm.remove("pongal",50);----  
  hm.get("poori");

GENERICS    
 why we learn generics ?  
 generics add stability to your code by  
 makking more of youre bugs detectable at compile time  
  }