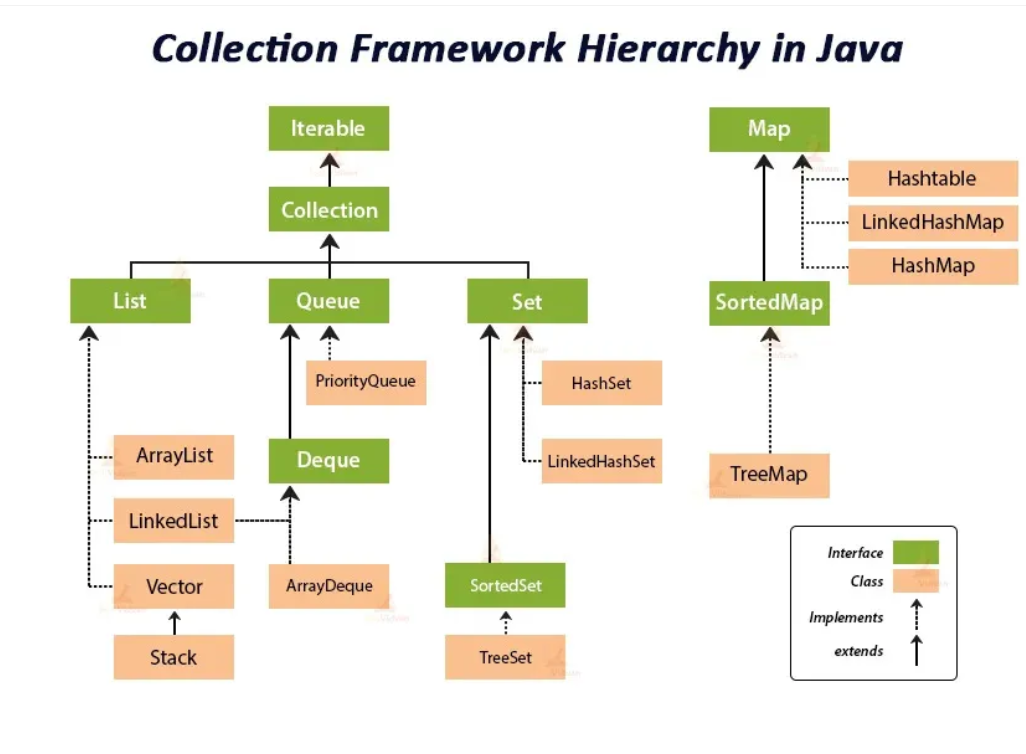
**Collection Framework | ArrayList, LinkedList & Collection Interface**

* Java Collection Framework
* Java Collection Interface
* Java List Interface
* Java Array List
* Java LinkedList
* Java stack

1. **Java Collection Framework :**

* **The java collection framework provides a set of interface and classes to implement various DSA.**



1. **Java Collection Interface**

* Collection interface is root interface of java collections framework which represents a group of objects know as elements
* Collection interface is a part of java.util package

Methods in collection interface- add(),remove, contains(check elements),size(),clear(remove all elements), isEmpty(check empty or not),add All(we merge two list),remove all(intersection),clear(list empty) ,toarray(object to array)

**Iterator**-

* iterator in java is an object that allows you to traverse through collection (like Arraylist, Hashset)
* it provides methods to access and remove elements during iteration
*  process of repeatedly accessing each element in a collection or a sequence, one after another. This can be done using loops (e.g., for, while) or with an Iterator.

A computer code on a black background

Description automatically generated

**Why use an Iterator?**

* Safe removal of elements while traversing.
* Can be used with any type of collection.
* **List->**
* ordered collection (like dynamic array )
* Allow duplicates
* List is Interface and it extends to collection
* Array List ,Linked list, vector, Stack

1. **Array List** – Resizable array (automatically adjust their capacity), allow duplicates, random access, Array List can store **null** values. Due to adding removing TC is O(n)

**Array List <String, Integer> AL=new Array List<>()**;

**Internal working in ArrayList ( new size=oldsize\*3/2+1) e**.g 10->increase by 6 ->10\*3->30/2=15+1=16

**Collection vs Collections**

* Collection is an interface that represents group of Objects
* Collections is an utility class that provides method to operate collection type(like searching ,sorting, modifying)

1. **Linked List:- Linked list is class implementation of List and Deque interface its doubly linked list List<> ls=new Linkedlist<>();**

* Linked list store elements in separate containers called nodes where each contain element and like to next node
* No need to resize like Array list
* A screenshot of a computer

  Description automatically generatedFaster when adding and removing elements
* Same method and programmatic working as Array List

1. **Vector :- vector is resizable array implementation of list interface (Vector<E> vc=new Vector<>());**

* the vector class synchronizes each individual operation. Means whenever we want to perform some operation on vector ,the vector class automatically applies a lock to that operation.
* When one thread is accessing a vector and at same time another thread want to access it get an exception called ConcurrentModificationException
* array List is not synchronized

1. **A screen shot of a computer code

   Description automatically generatedStack:- stack is subclass of vector that implements a LIFO data Structure .it provides method push(),pop(),peek(),empty()**

**A grid of text on a white surface

Description automatically generatedDifference between array list ,Linked list, Vector, Stack**

**Java Queue Interface**

The Queue is interface of the java collection framework provides the functionality of the queue Data structure .it extends collection interface(FIFO)

A screenshot of a computer screen

Description automatically generatedA blue squares with black text

Description automatically generated

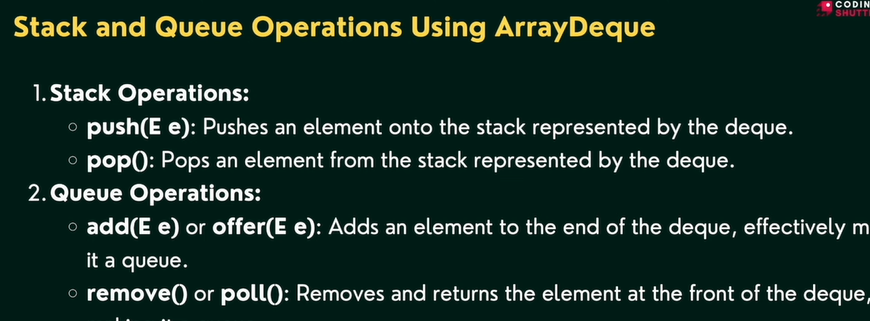
|  |  |
| --- | --- |
| * **offer(E e)🡪**insert elements return true false ( returns false if queue is full) | Boolean Add(E e)🡪insert element but return exception if queue is full |
| * **E poll()🡪** remove elements from queue and return null if q is empty | E remove 🡪remove but return exception if queue is empty |
| * **E peek()🡪** retrieve element which will be remove only showing and return null if queue empty | E element🡪 retrieve element if q empty throw an exception |

1. A screen shot of a computer code

   Description automatically generated**Linked List Queue**🡪

* The linked list class implements to queue Interface and provide method like peek(),poll(),offer()
* It follow FIFO

1. **Array Deque**🡪

* array Deque is resizable array based implementation of the Deque(Double ended queue)
* it allows elements to be added or removal from both side of queue (pollFirst(),peekFirst(),offerFirst(),getFirst)

A screen shot of a computer code

Description automatically generated

1. **Priority queue**🡪

* According to priority elements are remove and insert
* Elements which is higher priority are removed first if two elements have same priority they are removed in the ordered they were added
* React like minheap means always remove smallest element but we can also custom it like maxheap
* Automatically ordered from smallest element ascending

A screen shot of a computer code

Description automatically generated

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**Sets🡪**

**Java Set interface**

* **The set interface of java collection framework provides the features of the mathematical set in java**
* **It Extend the collection interface it cannot contains duplicate elements**
* **Time complexity is good o(n)**

A diagram of a set

Description automatically generated

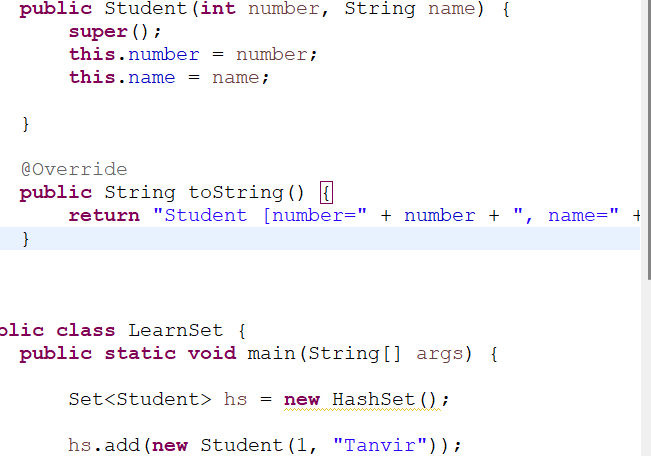
**Java Set Interface**

* Add()🡪add element to set
* addAll()🡪adds all element of the specified collection to the set
* remove()🡪remove specified elements from set
* remove all ()🡪remove all the elements
* retain all()🡪retains all the elements //intersection
* clear()🡪remove all the elements from set
* size()🡪return length of set
* contains🡪check elements pre

**Java Hash set**🡪

* in java hash set is commonly used if we have to access elements randomly
* hash set cannot contain duplicate hench each hash set has a unique hash code
* A screenshot of a computer code

  Description automatically generatedHash code is function like hash code(input)🡪output //hash code has unique identity

**Hash set Using custom objects**

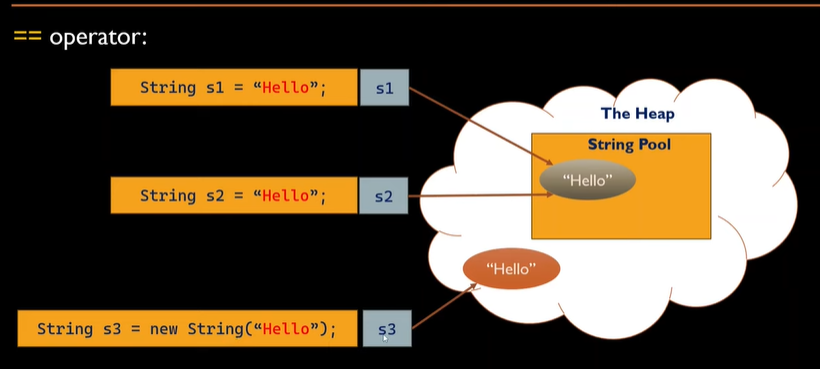
**Hash code and Equals Methods**

**In java both method is used for compare objects like equal or not**

**Equals() and == operator**

**“== Operator”**

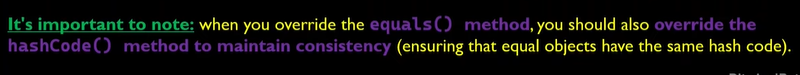
It compares the reference equality of two objects its check two objects references points to the same memory location or same object like below ..

****

* here for non primitive(object) we can see reference (s1,s2) are refer to same String Object(hello) and in same memory location so output is True
* for primitive types (int float, char) it compares actual value of variables

A computer screen shot of a program

Description automatically generated

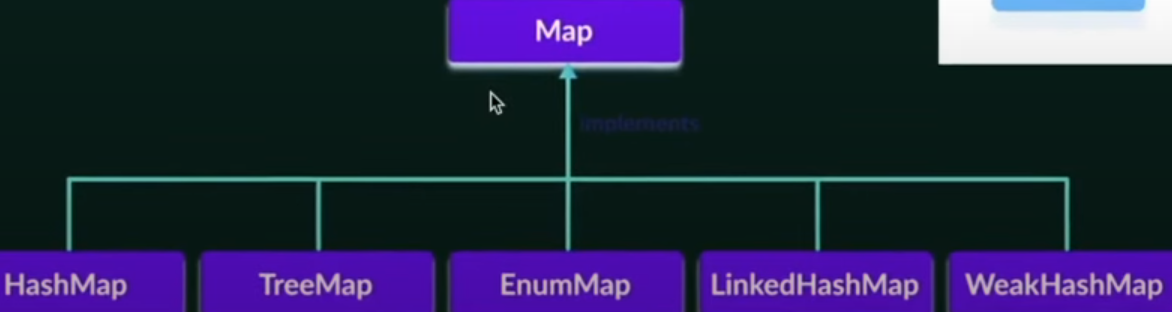


Hashcode() method🡪

* returns int value
* In Java, hashCode() and equals() are two important methods for comparing objects, especially when working with collections like HashMap, HashSet, or Hashtable. They serve distinct but related purposes.
* the hashcode method in java return a hash code value for the object
* it is used by hash-based data structures such as HashMap hash set
* we can create by eclipse

**Map interface🡪**

* In java elements of map are stored in key/value pairs. Keys are unique value associated with individual value
* A diagram of values

  Description automatically generatedA map cannot contains duplicate keys and each key is associated with single value
* Put (k, v)🡪for insert element with key value pair
* PutAll()🡪insert all entries from from the specified map to this map
* PutifAbset(k, v)🡪if absent keypair added
* Get(value or key)🡪get values
* getorDefault(“key” defaultvalue)🡪if value is not present then get defaultvalue
* containsKey(k)🡪checks the specified key is present or not
* containsValue()🡪to check value
* remove()🡪to remove key and using this we can remove key pair
* keyset()🡪we get set of key\
* values()🡪we get set of value
* entrySet()🡪all key and value in present in map
* all complexity O(n)

A screenshot of a computer program

Description automatically generated

**Comparable and comparator :🡪**

* To sort the custom object like student or employee, we need to provide explicit sorting logic
* To achieve this ,java provides the comparable and comparator interfaces.comparable and comparator interfaces in java allow us to define custom sorting behavior for objects,including sorting based on multiple data member