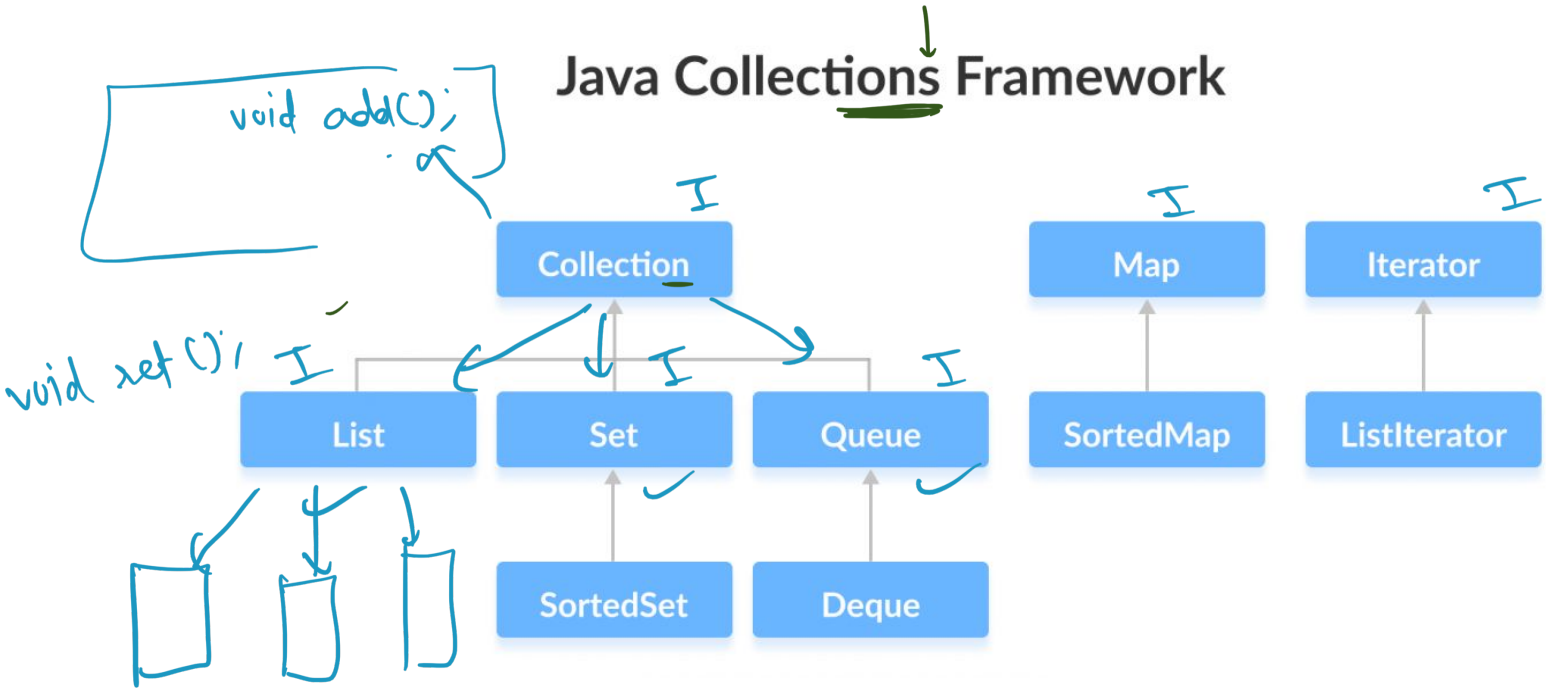
Collection Framework

Java Collections Framework



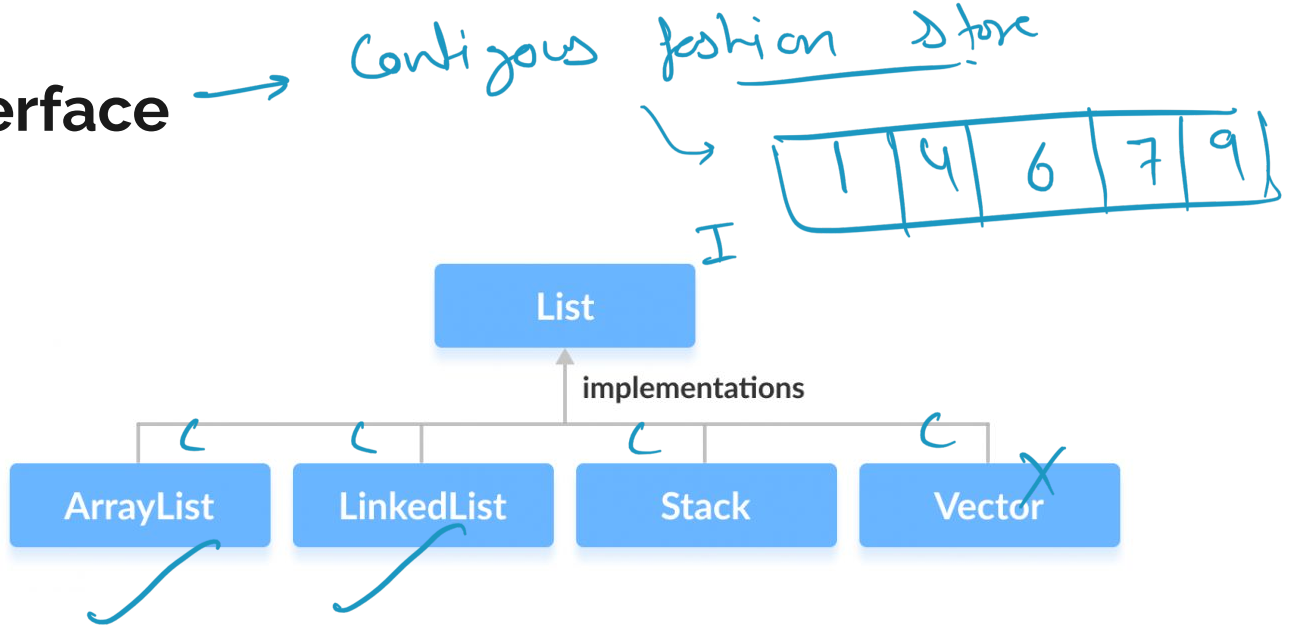
Abstraction in OOPS

Data Abstraction is the property by virtue of which only the essential details are displayed to the user. The trivial or the non-essential units are not displayed to the user. Ex: A car is viewed as a car rather than its individual components.

We can achieve Abstraction by these two methods:

1. Abstract Keyword
2. Interfaces

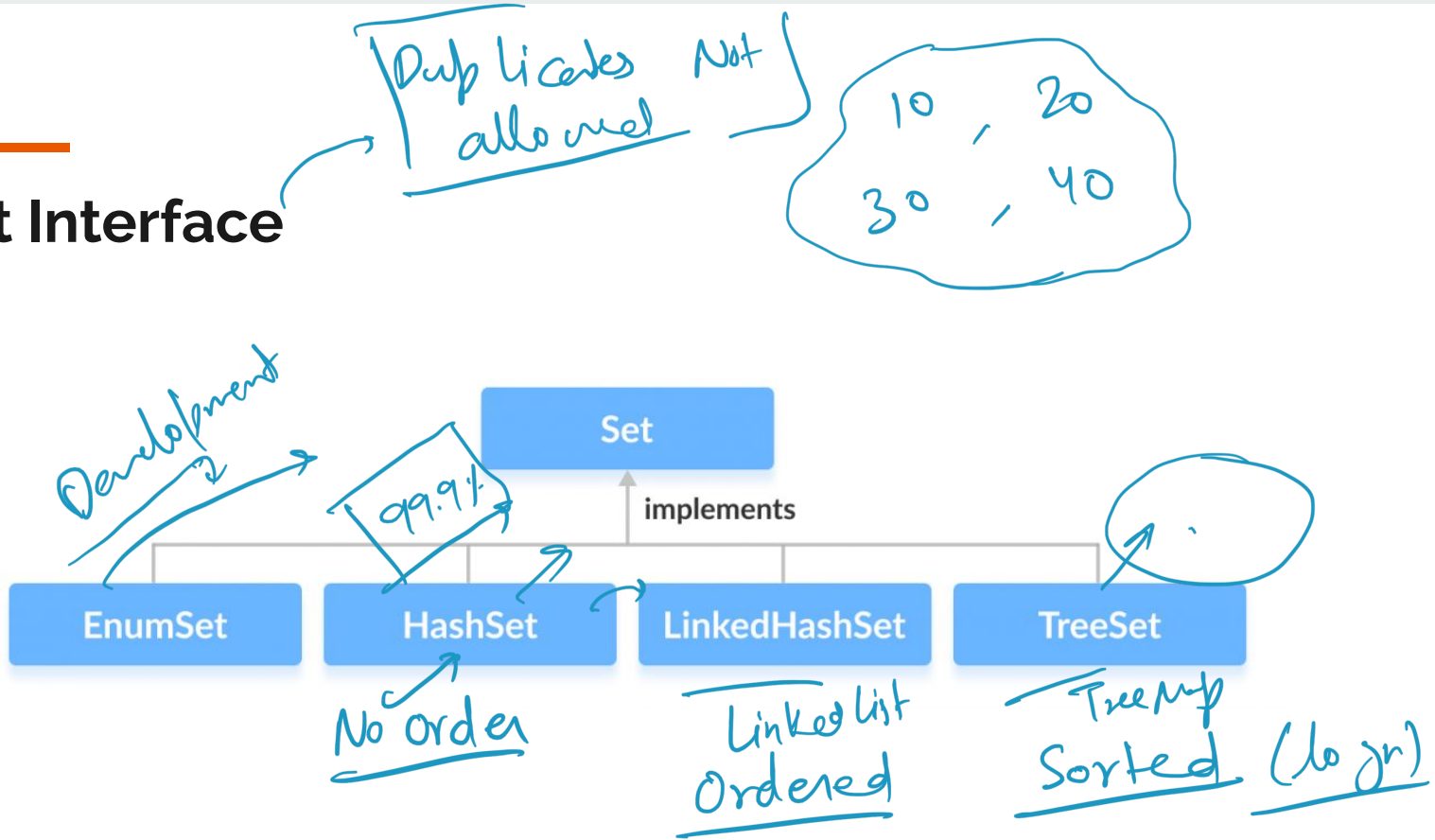
List Interface



Methods of List

- `add()` - adds an element to a list
- `addAll()` - adds all elements of one list to another
- `get()` - helps to randomly access elements from lists
- `iterator()` - returns iterator object that can be used to sequentially access elements of lists
- `set()` - changes elements of lists
- `remove()` - removes an element from the list
- `removeAll()` - removes all the elements from the list
- `clear()` - removes all the elements from the list (more efficient than `removeAll()`)
- `size()` - returns the length of lists
- `toArray()` - converts a list into an array
- `contains()` - returns `true` if a list contains specified element

Set Interface



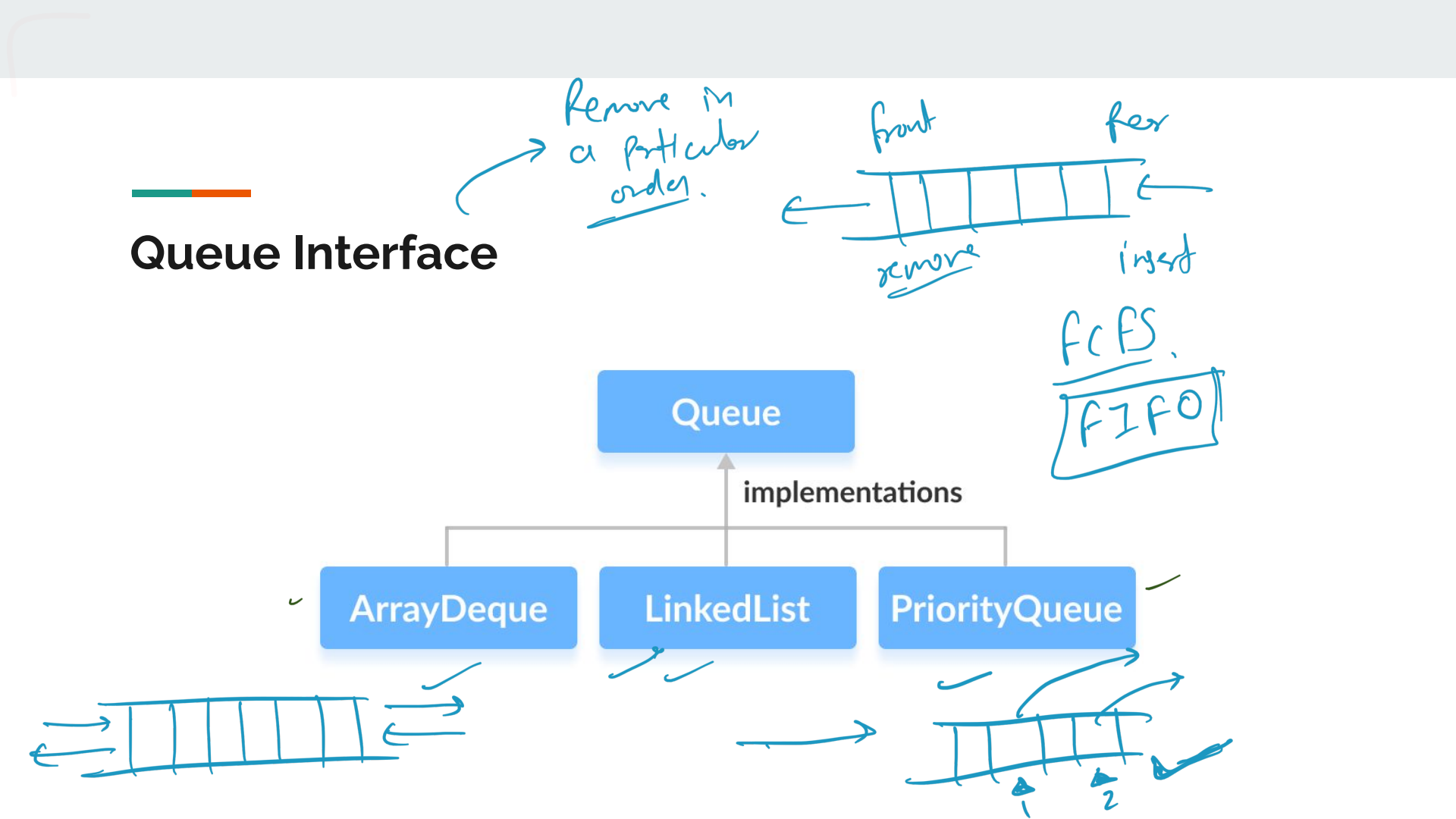
Methods of Set

A.removeAll(B)

- **add()** - adds the specified element to the set
- **addAll()** - adds all the elements of the specified collection to the set
- **iterator()** - returns an iterator that can be used to access elements of the set sequentially
- **remove()** - removes the specified element from the set
- **removeAll()** - removes all the elements from the set that is present in another specified set
- **retainAll()** - retains all the elements in the set that are also present in another specified set
- **clear()** - removes all the elements from the set
- **size()** - returns the length (number of elements) of the set
- **toArray()** - returns an array containing all the elements of the set
- **contains()** - returns `true` if the set contains the specified element
- **containsAll()** - returns `true` if the set contains all the elements of the specified collection
- **hashCode()** - returns a hash code value (address of the element in the set)

Object

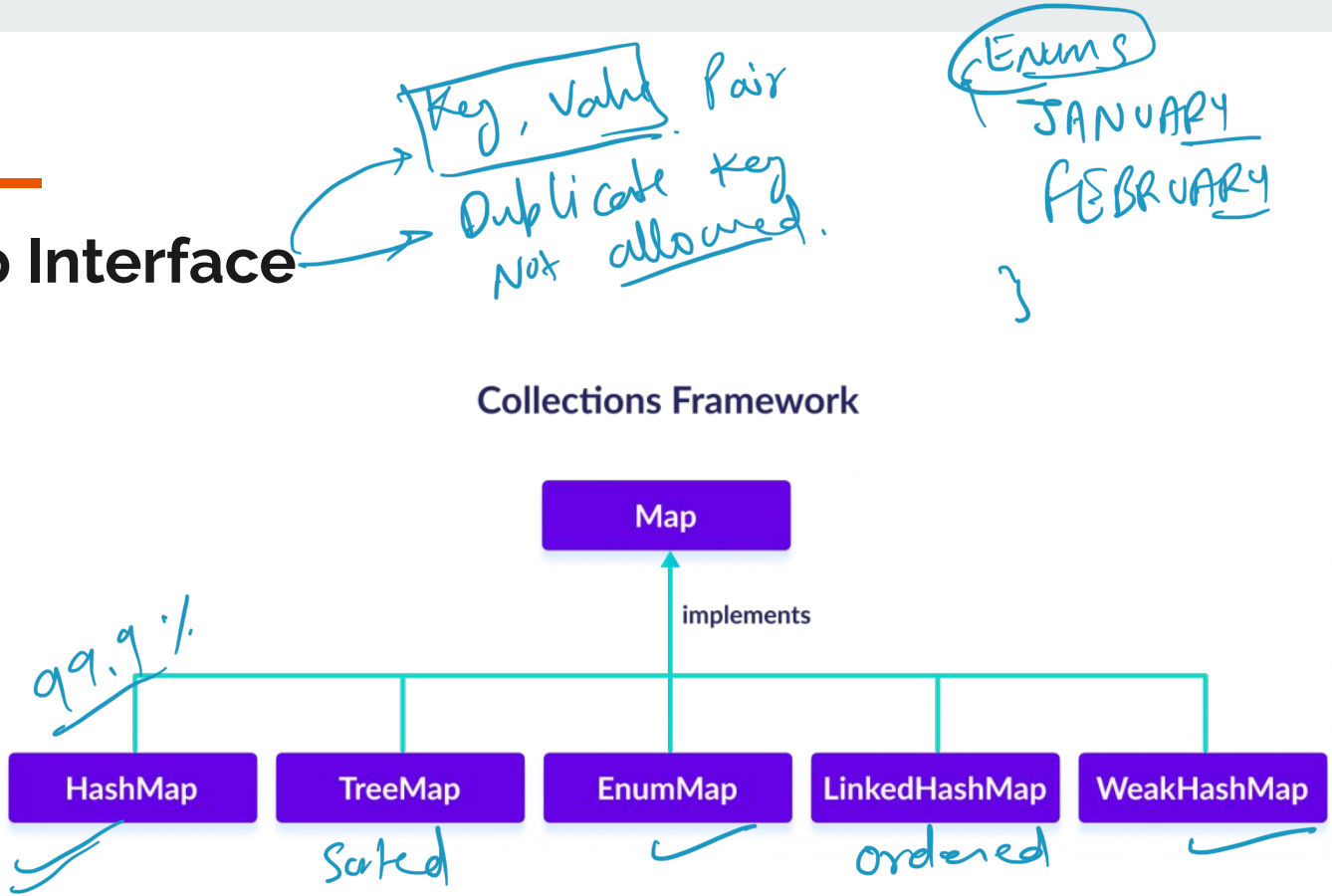
Queue Interface



Methods of Queue

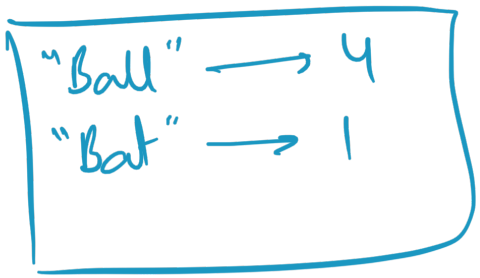
- **add()** - Inserts the specified element into the queue. If the task is successful, `add()` returns `true`, if not it throws an exception.
- **offer()** - Inserts the specified element into the queue. If the task is successful, `offer()` returns `true`, if not it returns `false`.
- **element()** - Returns the head of the queue. Throws an exception if the queue is empty.
- **peek()** - Returns the head of the queue. Returns `null` if the queue is empty.
- **remove()** - Returns and removes the head of the queue. Throws an exception if the queue is empty.
- **poll()** - Returns and removes the head of the queue. Returns `null` if the queue is empty.

Map Interface



Methods of Map

- **put(K, V)** - Inserts the association of a key `K` and a value `V` into the map. If the key is already present, the new value replaces the old value.
- **putAll()** - Inserts all the entries from the specified map to this map.
- **putIfAbsent(K, V)** - Inserts the association if the key `K` is not already associated with the value `V`.
- **get(K)** - Returns the value associated with the specified key `K`. If the key is not found, it returns `null`.
- **getOrDefault(K, defaultValue)** - Returns the value associated with the specified key `K`. If the key is not found, it returns the `defaultValue`.
- **containsKey(K)** - Checks if the specified key `K` is present in the map or not.
- **containsValue(V)** - Checks if the specified value `V` is present in the map or not.
- **replace(K, V)** - Replace the value of the key `K` with the new specified value `V`.
- **replace(K, oldValue, newValue)** - Replaces the value of the key `K` with the new value `newValue` only if the key `K` is associated with the value `oldValue`.
- **remove(K)** - Removes the entry from the map represented by the key `K`.
- **remove(K, V)** - Removes the entry from the map that has key `K` associated with value `V`.
- **keySet()** - Returns a set of all the keys present in a map.
- **values()** - Returns a set of all the values present in a map.
- **entrySet()** - Returns a set of all the key/value mapping present in a map.



String → Integer
↑ key
↑ value

Never cache!

Arrays class

Collections class