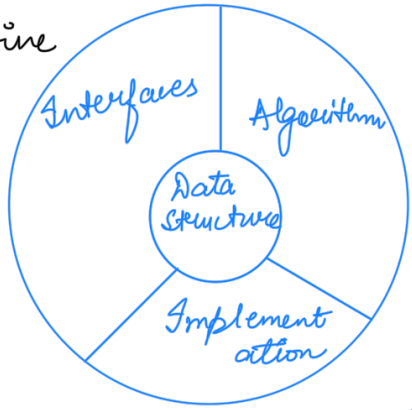
Java Collections Framework

→ Interfaces - Abstract types that define methods available on a collection like List, Set and Map.

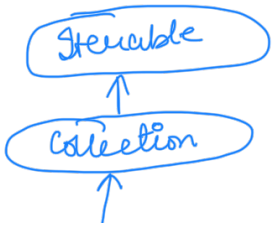


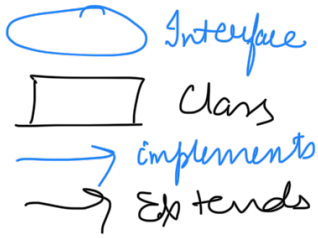
→ Implementation
→ Algorithm

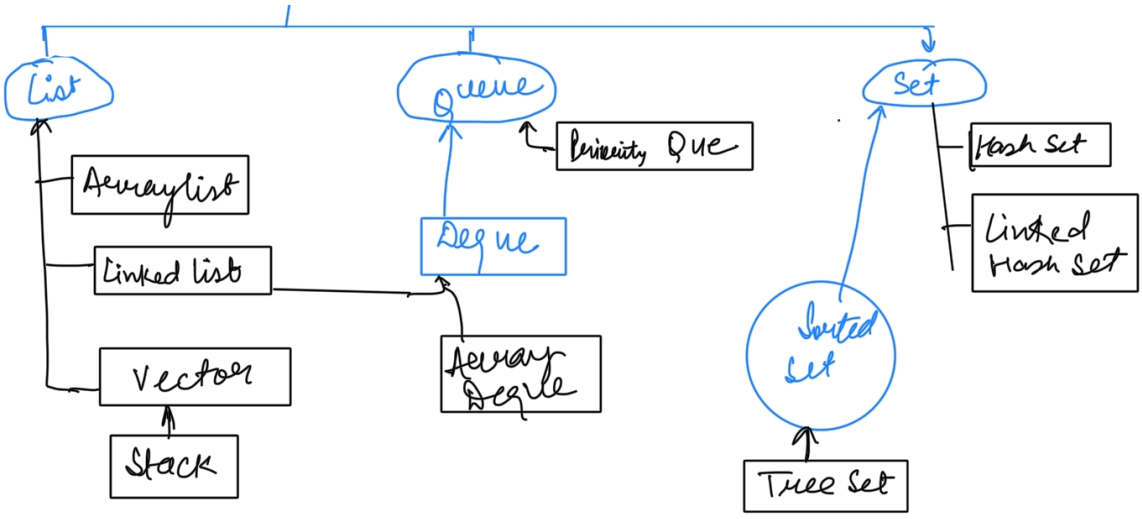
★ Implementation of Data Structures

List: ArrayList, LinkedList, Vector, Stack → Legacy class Thread Safe class Not to be used in single Threaded environ.
Set: HashSet, TreeSet, LinkedHashSet
Queue: LinkedList, ArrayDeque, PriorityQueue
Dequeue: LinkedList, ArrayDeque
Map: HashMap, TreeMap, LinkedHashMap

★ Collection Class (Implementation of basic Algo)
binarySearch(), sort, min etc.







✱ Generics

- Write once use for any non-primitive data type
- Java collection use Generics
- Generic class / Interface and method.
- Type safety.

✱ `Object[] toArray()`
`T[] toArray(T[])`

✱ Immutable class

→ Declare class as final so that it can't be extended.
→ Data Members as private so that they are not accessible
→ Data Members should be final so that value can't be changed after object creation.
→ A parameterized constructor should initialize fields doing deep copy to stop modification of data members with object reference.
→ Deep copy of object in getter methods to return copy rather than returning object reference

✱
```
q.containsAll(q2);  // return true if q contains all elements of q2 otherwise false
q.retainAll(q2);    // retains elements of q2 in q and removes other
t.removeAll(t2);    // removes all elements of t2 from t.
s.removeIf((n) -> (n%2==0))
```

Java Streams

→ Stream function is present in the collection Interface. All