Java Collections

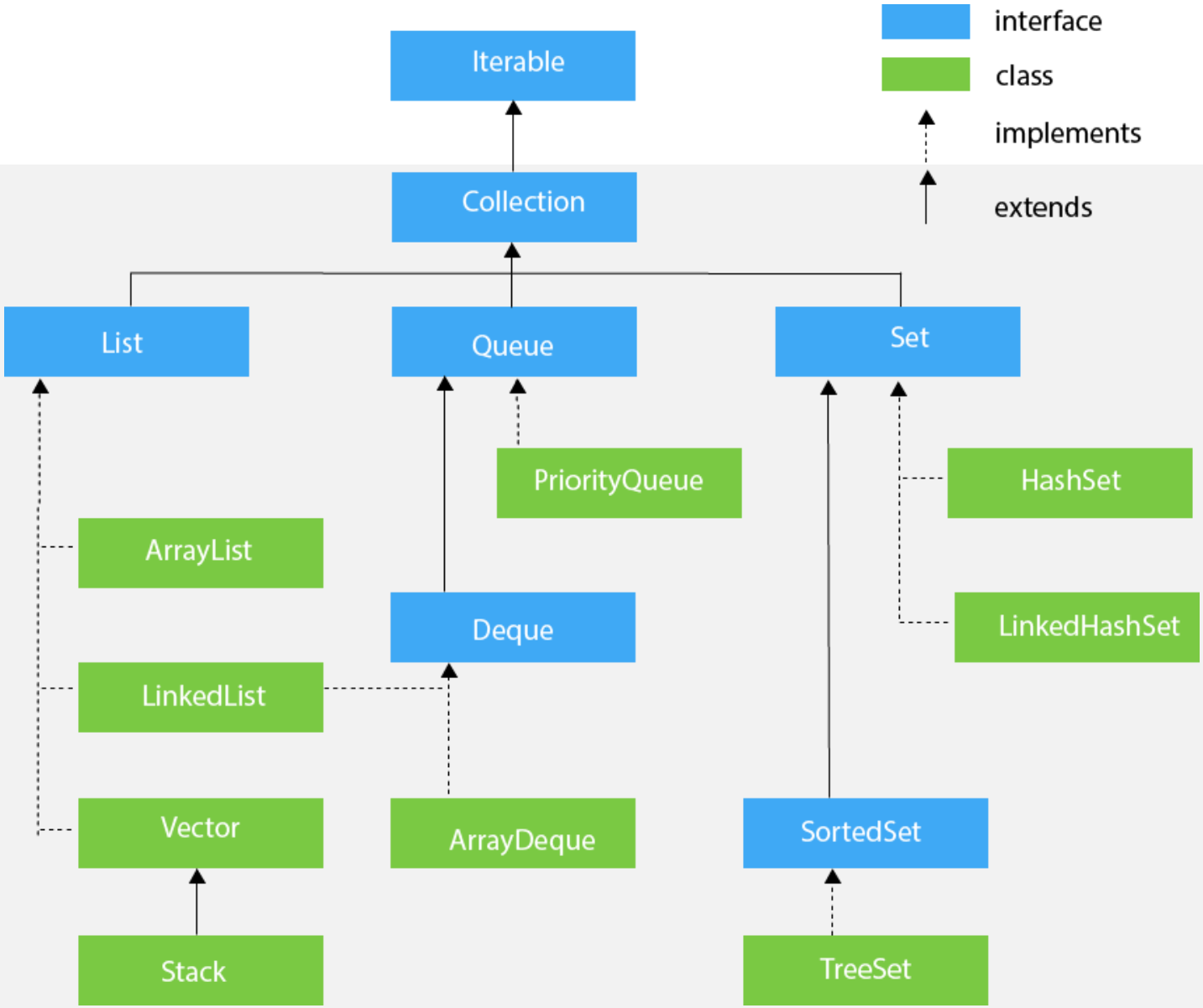
Archer Infotech , PUNE

Java Collections

What is Collections Framework ?

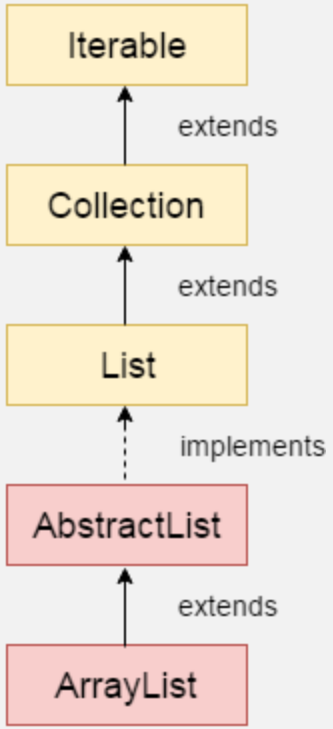
- **The Java Collections Framework is a collection of interfaces and classes which helps in storing and processing the data efficiently**

- .


interface
class
implements
extends
Iterable
Collection
List
Queue
Set
PriorityQueue
HashSet
ArrayList
Deque
LinkedHashSet
LinkedList
Vector
ArrayDeque
SortedSet
Stack
TreeSet

ArrayList

- Java ArrayList class can contain duplicate elements.
- Java ArrayList class maintains insertion order.
- Java ArrayList class is non synchronized.
- Java ArrayList allows random access because array works at the index basis.
- In ArrayList, manipulation is 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.



Creating ArrayList

- ArrayList list=**new** ArrayList();//creating old no n-generic arraylist

- ArrayList<String> list=**new** ArrayList<String>(); //creating new generic arraylist

Iterating ArrayList

- Iterating ArrayList using Iterator

```
Iterator itr=list.iterator
 while(itr.hasNext ) {
            System.out.println(itr.next())
}
```

- Iterating ArrayList using For-each loop

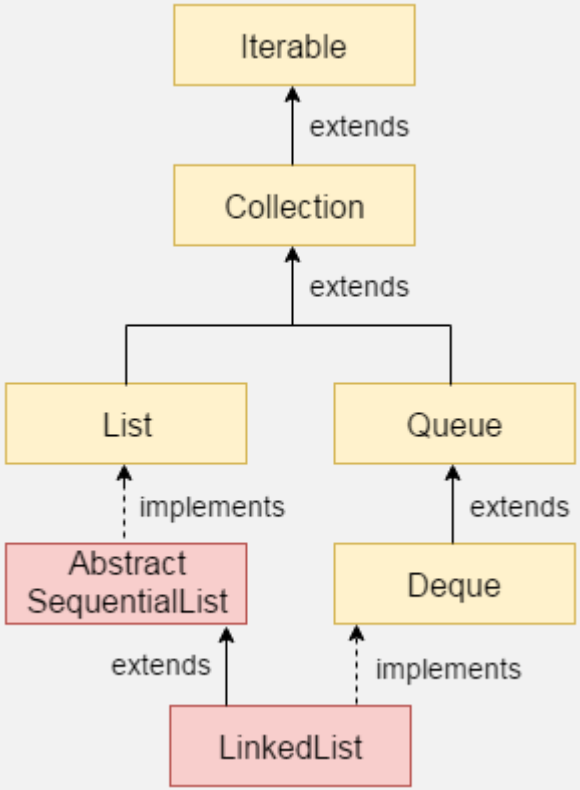
```
for(String fruit:list)
{
            System.out.println(fruit);
}
```

Sorting ArrayList

```
1.//Sorting the list
2.  Collections.sort(list1);
```

Linked List

- 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.



Stack

The stack is the subclass of Vector. It implements the last-in-first-out data structure, i.e., Stack. The stack contains all of the methods of Vector class and also provides its methods like boolean push(), boolean peek(), boolean push(object o), which defines its properties.

Queue Interface

Queue interface maintains the first-in-first-out order.

```
1.Queue<String> q1 = new PriorityQueue();
2.Queue<String> q2 = new ArrayDeque();
```

Priority Queue

```
PriorityQueue<String> queue=new PriorityQueue<String>();
queue.add("Amit Sharma");
queue.add("Vijay Raj");
queue.add("JaiShankar");
queue.add("Raj");
System.out.println("head:"+queue.element());
System.out.println("head:"+queue.peek());
System.out.println("iterating the queue elements:");
Iterator itr=queue.iterator();
while(itr.hasNext()){
System.out.println(itr.next());
}
queue.remove();
queue.poll();
System.out.println("after removing two elements:");
Iterator<String> itr2=queue.iterator();
while(itr2.hasNext()){
System.out.println(itr2.next());
1.}
```

THANK YOU !!!