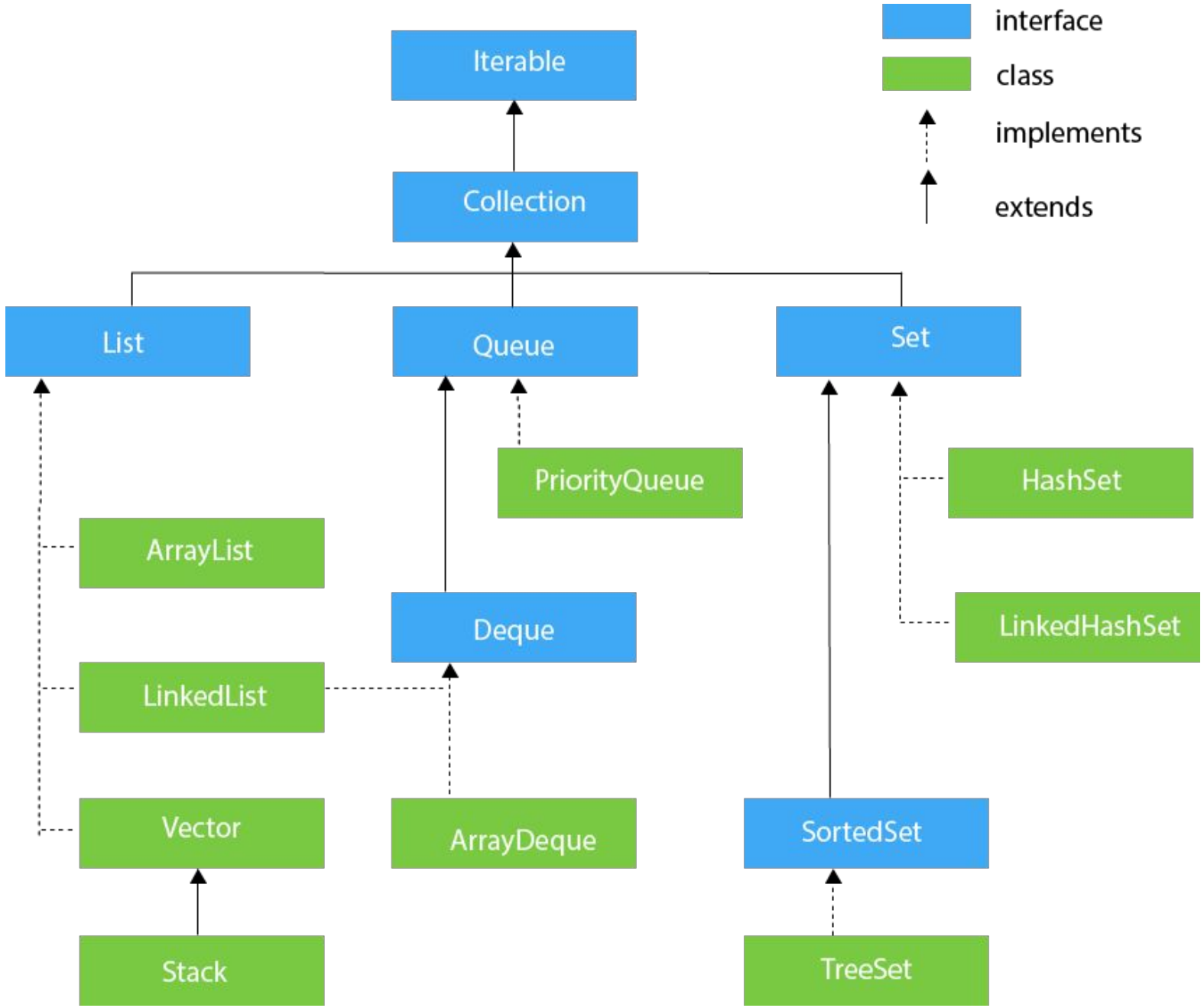
Data structures and Algorithms

By

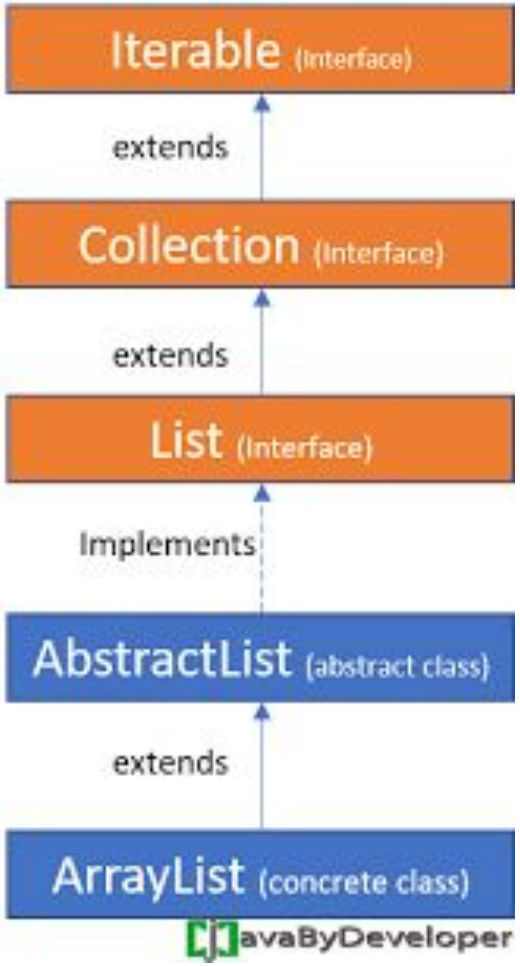
Engr Fatima Jaffar

Collection Framework

- A java collection framework provides an architecture to store and manipulate a group of objects, which includes classes, interfaces and algorithms


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

Hierarchy of Arraylist

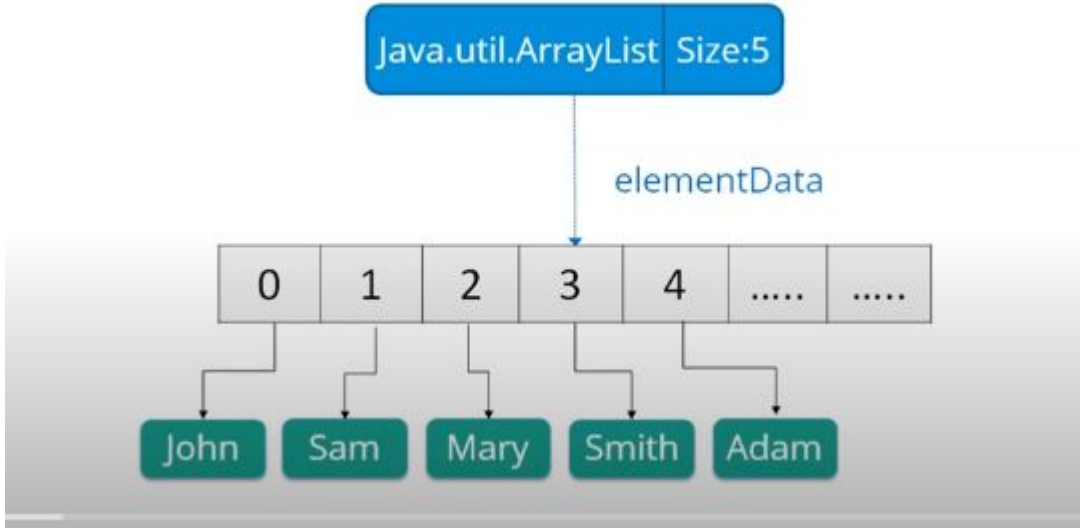


Arraylist?

- Uses dynamic array for storing the elements.
- It inherits Abstract List class and implements list interface.

Arraylist?

- It is a part of Collection framework and implementation of List interface through which we can add elements dynamically.
- Size of the array increased dynamically if size of input increases than the initialized size.

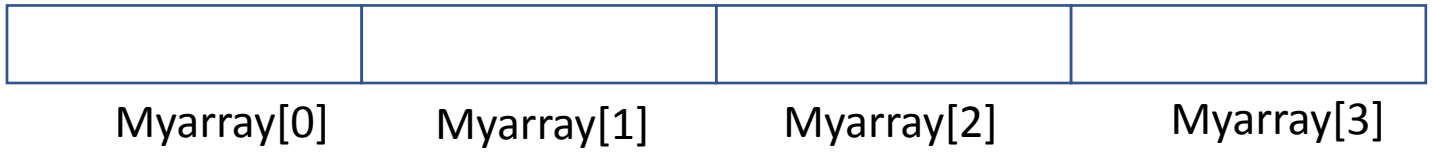


ArrayList

- Arraylist is initialized by a size but if an item is added or removed, arraylist automatically adjusts the item by expanding or shrinking the actual size.
- Arraylist can not be used for primitive data types like int,char,bool,float.
- To access data types we use wrapper classes.

Internal working of Arraylist

Array List Initialization

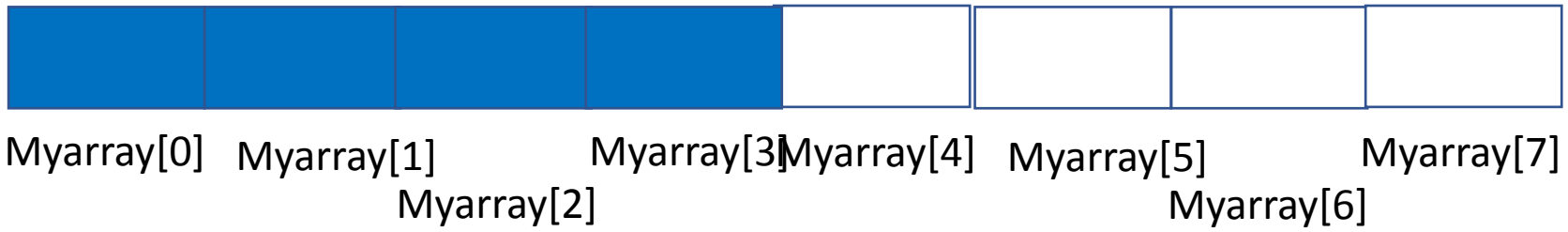


Adding elements

Internal working of Arraylist

Capacity of array is full

Size of array is doubled



Constructors in Arraylist

- 1) Arraylist()

  - Builds an empty arraylist
  - Syntax:

    ArrayList<E> myarray=new ArrayList<E>();

- ArrayList(Collection c)

- This constructor is used to add all the elements of specified collection c to the arraylist
- Syntax:

  public Boolean addAll(Collection c)

- ArrayList(int capacity)
- Built with specified initial capacity(number of cells)
- Syntax:

  ArrayList myarray=new ArrayList(int initial capacity)

add(Object o)	This method is used to append a specific element to the end of a list.
addAll(Collection C)	This method is used to append all the elements from a specific collection to the end of the mentioned list, in such an order that the values are returned by the specified collection's iterator.
addAll(int index, Collection C)	Used to insert all of the elements starting at the specified position from a specific collection into the mentioned list.
clear()	This method is used to remove all the elements from any list.
clone()	This method is used to return a shallow copy of an ArrayList.
contains?(Object o)	Returns true if this list contains the specified element.
ensureCapacity?(int minCapacity)	Increases the capacity of this ArrayList instance, if necessary, to ensure that it can hold at least the number of elements specified by the minimum capacity argument.
forEach?(Consumer<? super E> action)	Performs the given action for each element of the Iterable until all elements have been processed or the action throws an exception.
get?(int index)	Returns the element at the specified position in this list.
indexOf(Object O)	The index the first occurrence of a specific element is either returned, or -1 in case the element is not in the list.
isEmpty?()	Returns true if this list contains no elements.

```java
// Java Program to Add elements to An ArrayList

// Importing all utility classes
import java.util.*;

// Main class
class Test_array_list {

    // Main driver method
    public static void main(String args[])
    {
        // Creating an Array of string type
        ArrayList<String> al = new ArrayList<>();

        // Adding elements to ArrayList
        // Custom inputs
        al.add("Hello");
        al.add("World");

        // Here we are mentioning the index
        // at which it is to be added
        al.add(1, "new");

        // Printing all the elements in an ArrayList
        System.out.println(al);
    }
}
```