Variable   -> hold data(only one data)
ArrayVaraible   ->10,20,30,40

Advnatage of Array
----
   1. Large volume of data can be holded by single varaible
   2. To access the data we use "index".

DisAdvantage of Array
----
   1. It expects continous memory
   2. Once Array is fixed, u can't increase or decrease the size based on our requirement.
   3. It can hold only homogenous data.
   4. Array is not implemented based on any standard datastructure,so ready made methods are not avaiable
       to perform some operation like sorting,searching,etc....It increase the burden on developers to write
        their own logic to perform these common operations.
                  int a[5]= { 10,20,0,5,2,1};
                        Arrays.sort(a);

To overcome all the limitations of an array we use "Collections"
  Advantage of Collections
      1. Collections are growable in nature,u can increase or decrease the size of an Collection.
       2. It can hold both homogenous and heterogenous Objects.
      3. All Collection(I)implementation classes are built by following some standard DataStructure.
      4. The data stored inside collection would always be in Object type only
                  primtive ---autoboxing----------> object

Collection(I)
       |
  List(I)

List(I)
1. If we want insertion order to be preserved and duplicates to be allowed
2. Index plays a vitol role here.
3. internally datastructure followed is "Array".

      eg: ArrayList,LinkedList, Vector,Stack

Set(I)
 1. Insertion order won't be preserved and duplicates should not be allowed.
 2. index won't play any vitoal role
 3. internally datastrucure follwed is "Hash Table".

      eg: HashSet,LinkedhashSet

SortedSet(I)
 1. Inside Set, if the elements/Object has to be Sorted then we need to opt for SortedSet.
            eg: TreeSet

Note: For Collection interface the parent class is "Iterable"

Iterable -> The entity which can be used in iteration.

eg: loops

When to use List, Set?
List -> This implementaion classes should be used for the following conditions
a. insertion order and duplicates are allowed
eg: ArrayList, LinkedList

ArrayList -> We add the elements, we need to remove
The array size should be shrinked at the runtime(more time -> don't use)
If the frequent operation is read operation then we use ArrayList.
ArrayList implements an interface called "RandomAccess".
Note:ArrayList and Vector are the only 2 classes which implements "RandomAccess".

LinkedList-> We add the elements, we need to remove
linkedlist -> data will be stored in scattered manner not in continous mode.
deletion is easy,but reading is tough.
if the frequent operation is deletion then we use LinkedList

Marker Interface
An interface which does not contain any abstract methods that interface is called "Marker Interface".
If an class implements Marker interface, then that class "object" will get additional funcitionality at the runtime by the
JVM.
eg: RandomAccess(searching is fast in the Collection object), Serializable(makes the object transportable),.....

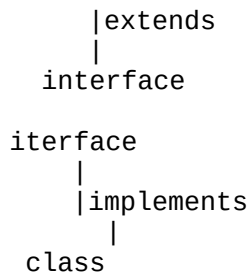
Set

Set
|
SortedSet(I)
| implements
TreeSet
=> Best suited to keep the elements in sorted order
=> Sorted order can be "Ascending/Descending" order.
=> 2 important interfaces
a . Comparable
b. Comparator

```
int a[5] = {10,20,30,40,50};
System.out.println(a[3]);// directly 40 printed by accessin at the memory level

ArrayList al =new ArrayList();
al.add(10);
al.add(20);
al.add(30);
System.out.println(al.get(2));//performance is low compared with Array.
```

Relation b/w 2 interface
iterface
|


|extends
|
interface
iterface
|
|implements
|
class