Collections

Array : 10, 20, 30, 40

Single unit value — Sri, null, ...

int a = 10;

String name = "Sri";

boolean b = true;

50 → 60 → 70

Disadvantage of arrays:

→ Size of the array is fixed during the creation

→ Need contiguous memory to store the elements

int a[] = new array[5]

RAM (heap area)

→ only homogeneous data can be stored

→ It is compulsory to know the no. of elements that we would be inserting into an array

a[0] = "nandu"

a[1] = true

a[2] = 'c'

→ a[5]

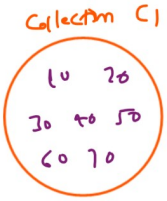
→ a[10]

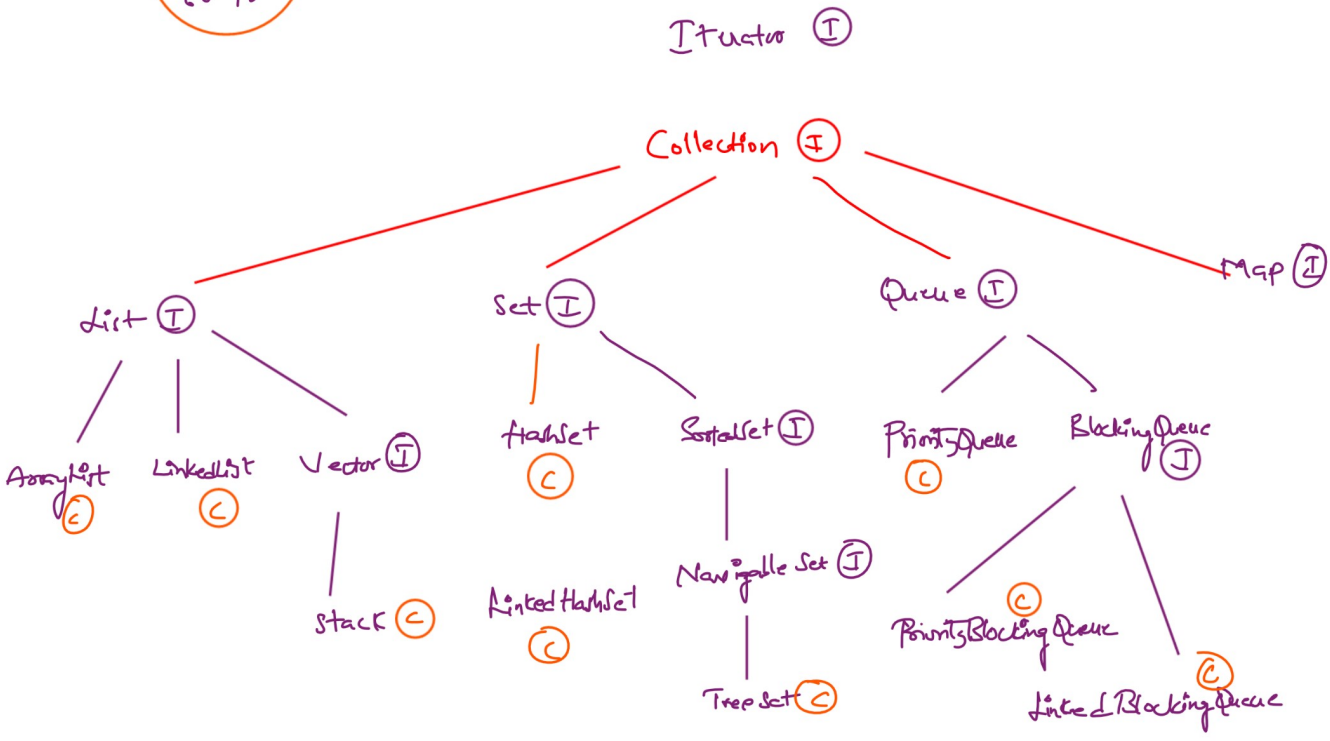
5

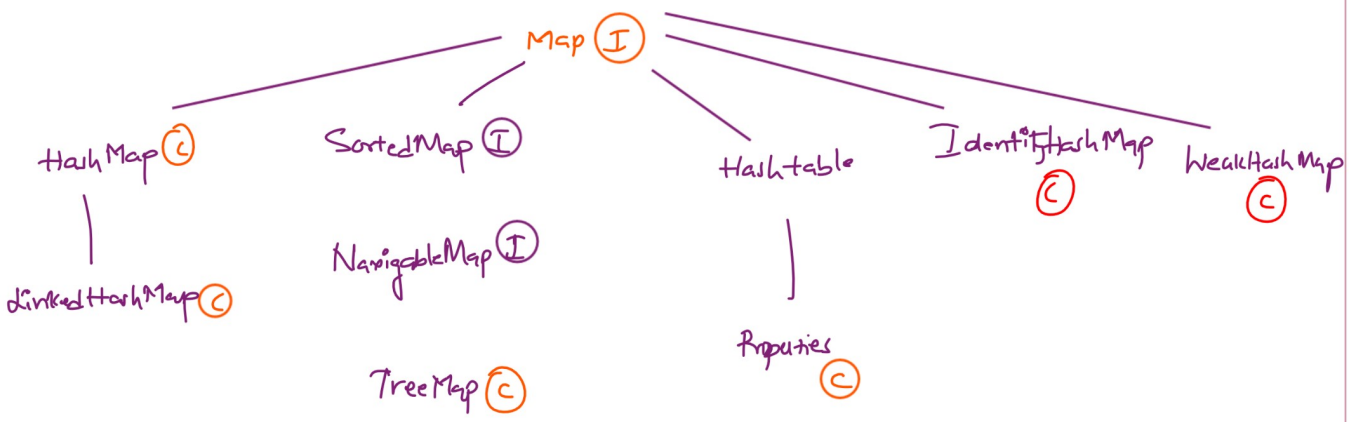
a[10]

a[5]

def: Collection is used, when we need to store multiple objects in a single entity

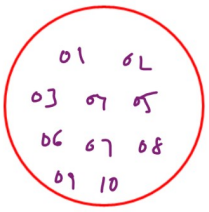




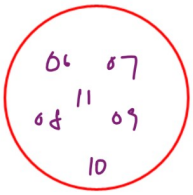


Common methods (in Collection)

Collection C1



Collection C2



return type	
boolean	add (Object O)
	addAll (Collection C)
	remove (Object O)
	removeAll (Collection C)
	retainAll (Collection C)
void	clear()
int	size()
Iterator	iterator()
Object[]	toArray()
boolean	contains (Object O)
	containsAll (Collection C)
	isEmpty()

Note:

C1.removeAll(C2)

→ the above statement, would remove all objects from C1 which are present in C2

i.e if remove 06, 07, 08, 09, 10 objects from collection C1

→ C1.retainAll(C2)

C1 retains (keep) all the elements present C2 & removes other elements in C1

→ i.e 01, 02, 03, 04, 05 gets removed from collection C1

List(I)

Properties

→ insertion ordered is maintained

→ NULL insertion is allowed

→ indexing is maintained

→ duplicates are allowed

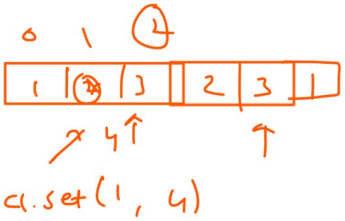
Methods of List:

boolean
- add(int index, Object o)
- addAll(int index, Collection c)
- remove(int index)
- set(int index, Object o)

int — indexOf(Object o)

int — lastIndexOf(Object o)

ListIterator — listIterator()



c1.set(1, 4)