

Java Collection Framework

Array drawback:

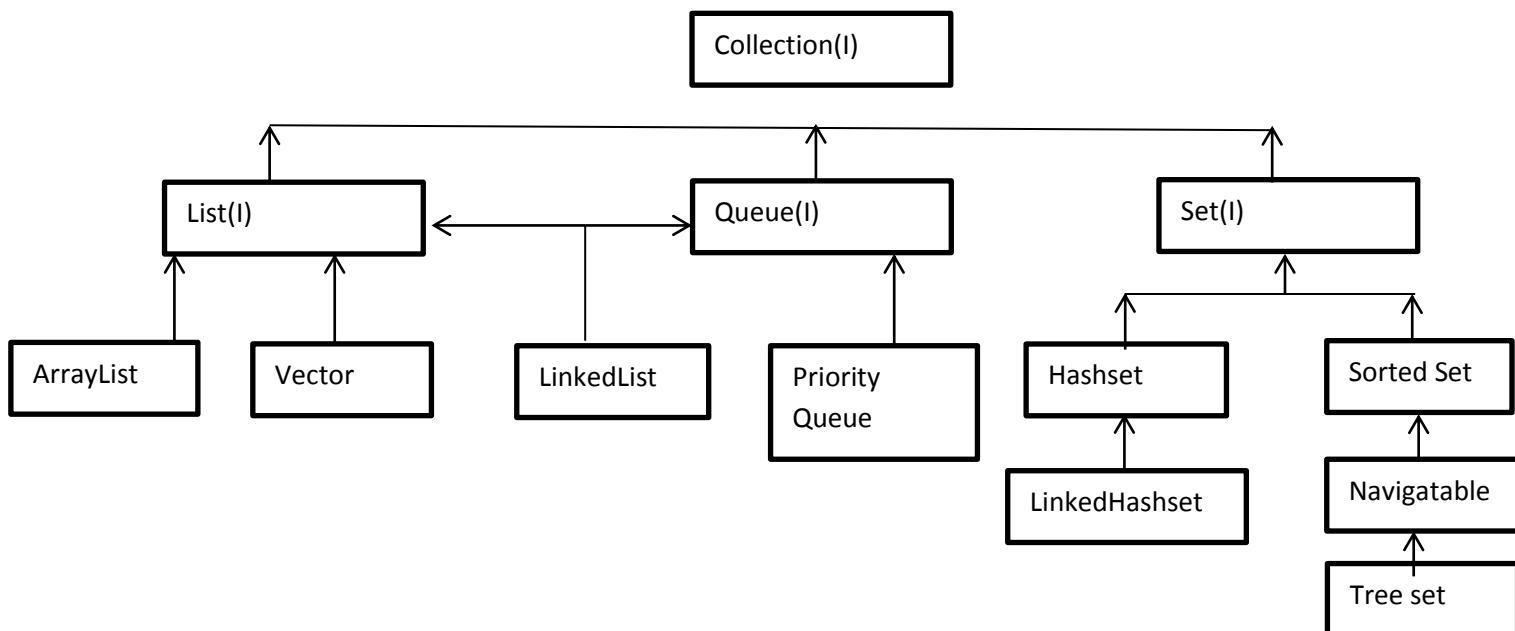
1. Array doesn't allow us to store heterogeneous elements.
2. Array size is fixed and we cannot change array size during runtime.
3. In array there are no built in function to perform operations.

Java collection framework is a set of classes and interface defined in java.util package. Classes and interfaces are used to manage group of objects.

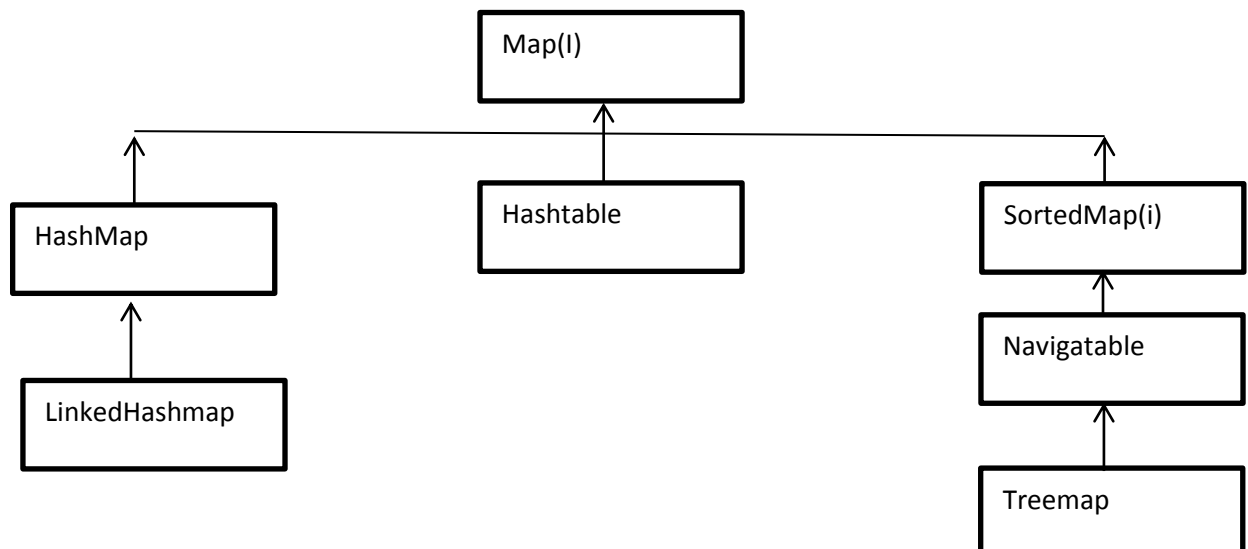
Collection is an entity used to store the group of objects. Collection allow to store heterogeneous elements.

Collection allows to store only objects and if we try to insert primitive data type, the elements will be converted into objects using wrapper class. Collection size is not fixed, it grows dynamically at runtime.

JCF hierarchy:



Map Hierarchy



The interface and classes of JCF are defined in java.util package.

Collection method:

1. public Boolean add(Object ob)

List

List is a child interface of collection . Its defined in java.util package.
List arrange elements in index format, hence it is also called as index based collection.
There are 3 implementation classes of list interface:

1. ArrayList
2. Linkedlist
3. Vector

List allows to store null value, duplicate elements and heterogeneous elements.

ArrayList

ArrayList is an implementation class of List interface. Its defined in java.util package. ArrayList is an index based collection, it allows to store null values, duplicate elements and heterogeneous elements. ArrayList preserve the order of insertion. Default capacity of an arraylist is 10 and grows size dynamically using formula:
New capacity= $(cc*3/2)+1$

On growing dynamically, old arraylist is copied into new arraylist

```
ArrayList l=new ArrayList();
```

```
l.add(10);  
l.add(20);  
l.add(30)  
.  
.  
.  
.  
l.add(100);  
l.add(110);
```

10	20	30	40	50	60	70	80	90	100
----	----	----	----	----	----	----	----	----	-----

NC= $(10*3/2)+1$
=16

10	20	30	40	50	60	70	80	90	100	110				
----	----	----	----	----	----	----	----	----	-----	-----	--	--	--	--

1. Program to show add function using arraylist:

```
public static void main(String[] args)
{
    ArrayList a=new ArrayList();
    a.add(10);
    a.add("Java");
    a.add(2.4);
    a.add('a');
    a.add(null);
    a.add("Java");

    System.out.println(a.toString());
    System.out.println(a); // toString is called by default
    System.out.println(a.size());
    Object o=a.get(1); // return value of index mentioned
    System.out.println(o);

    for (int i = 0; i < a.size(); i++)
    {
        Object o1=a.get(i);
        System.out.println(o1);
    }
}
```

2. Program to use methods of list and collection

```
public static void main(String[] args)
{
    ArrayList a=new ArrayList();
    a.add(10);
    a.add("Java");
    a.add(2.4);
    a.add('a');

    boolean b=a.isEmpty(); // verify whether arralist is empty
    System.out.println(b);
    boolean b1=a.contains("java"); // verify whether value is there in the list
    System.out.println(b1);
    System.out.println(a);
    a.remove(3);
    System.out.println(a);
}
```

3. Prog to display details of all students in the collection

```
public static void main(String[] args)
{
    Student s1=new Student(1,"Alex",60.0);
    Student s2=new Student(2,"Karev",50.0);
    Student s3=new Student(3,"John",65.0);
    Student s4=new Student(4,"jerry",80.0);
    Student s5=new Student(5,"Jenny",70.0);

    ArrayList a=new ArrayList();
    a.add(s1);
}
```

```

        a.add(s2);
        a.add(s3);
        a.add(s4);
        a.add(s5);

        for (int i = 0; i < a.size(); i++)
        {
            Object o=a.get(i);
            Student s=(Student)o;

            if((s.marks>=60)&&(s.marks<=70))
            {
                System.out.println(s.name + "\t" + s.id + "\t" +s.marks);
            }
        }
    }
}

```

5. Collections sort

```

public static void main(String[] args)
{
    ArrayList a=new ArrayList();
    a.add(20);
    a.add(30);
    a.add(10);
    a.add(40);

    Collections.sort(a);
    System.out.println(a);
}

```

6. WAP to store objects inside collection and display only details of employee name starting with A.

```

public static void main(String[] args)
{
    Employee e1=new Employee(1,"Alex",60000.0);
    Employee e2=new Employee(2,"Karev",50000.0);
    Employee e3=new Employee(3,"Adam",650000.0);
    Employee e4=new Employee(4,"jerry",80000.0);
    Employee e5=new Employee(5,"Ali",700000.0);

    ArrayList a=new ArrayList();
    a.add(e1);
    a.add(e2);
    a.add(e3);
    a.add(e4);
    a.add(e5);

    for (int i = 0; i < a.size(); i++)
    {
        Object o=a.get(i);
        Employee e=(Employee)o;
        boolean b= e.name.startsWith("A");
        if(b==true)
        {

```

```
        }  
    }  
}
```

```
System.out.println(e.name + "\t" + e.id + "\t" + e.salary);
```