**Cursors in Java Collection**

If we want to get objects one by one from the collection we should go for cursor.

 There are 3 types of cursors available in java

 1). **Enumeration**

 2). **Iterator**

 3). **List Iterator**

**Enumeration**

1). We can use Enumeration to get objects one by one from the legacy collection objects.

2). We can create Enumeration object by using elements() method.

public Enumeration elements();

Enumeration e=v.elements();

using Vector Object

Enumeration interface defines the following two methods

public boolean hasMoreElements();

public Object nextElement();

## Limitation of Enumeration

* We can apply the Enumeration concept only for legacy classes and it is not a universal cursor.
* By using Enumeration we can get only read access and we can't perform remove operations.
* To overcome these limitations sun people introduced Iterator concept in 1.2v.

**Iterators**

1. We can use Iterator to get objects one by one from any collection object.
2. We can apply Iterator concept to any collection object and it is a universal cursor.
3. While iterating the objects by Iterator we can perform both read and remove operations.
4. We can get Iterator object by using iterator() method of Collection interface.

public Iterator iterator();

Iterator itr = c.iterator();

The iterator interface defines the following 3 methods.

public boolean hasNext();

public object next();

public void remove();

**Limitations of Iterator**

* Both enumeration and Iterator are single direction cursors only. That is we can always move only forward direction and we can't move to the backward direction.
* While iterating by Iterator we can perform only read and remove operations and we can't perform replacement and addition of new objects.
* To overcome these limitations sun people introduced listIterator concept.

**ListIterator**

1). ListIterator is the child interface of the Iterator.

2). By using listIterator we can move either to the forward direction (or) to the backward direction that is it is a bi-directional cursor.

3). While iterating by listIterator we can perform replacement and addition of new objects in addition to read and remove operations.

4). By using listIterator method we can create listIterator object.

**NOTE** : The most powerful cursor is listIterator but its limitation is it is applicable only for "List objects".

