Cursors

Overview of Cursors

- If we want to get Objects One by One from the Collection then we should go for Cursors.
- There are 3 Types of Cursors Available in Java.
  1) Enumeration
  2) Iterator
  3) ListIterator

Enumeration

Overview

1. get Objects One by One from the Collection.
2. We can Create Enumeration Object by using elements()
   a. public Enumeration elements();
3. Eg:
   Vector v = new Vector();
   Enumeration e = v.elements();
4. Methods
   a. public boolean hasMoreElements();
   b. public Object nextElement();

Limitations of Enumeration:

- Enumeration Concept is Applicable Only for Legacy Classes.
- By using Enumeration we can Perform Read Operation and we can't Perform Remove
- Operation.

Iterator

Overview

- We can Use Iterator to get Objects One by One from Collection.
- We can Apply Iterator Concept for any Collection Object. Hence it is a Universal Cursor.
- By using Iterator we can Able to Perform Both Read and Remove Operations.
- Using Iterator we can Move Only towards Forward Direction.
- We can Create Iterator object by using the iterator() of Collection Interface.
  public Iterator iterator();

- eg.,
  Collection col = new ArrayList<>();
  Iterator itr = col.iterator();

Methods

1. public boolean hasNext()
2. public Object next()
3. public void remove()

Limitations

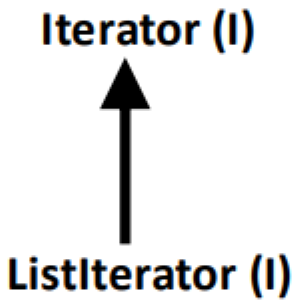
- By using Enumeration and Iterator we can Move Only towards Forward Direction and we can't Move Backward Direction.
- By using Iterator we can Perform Only Read and Remove Operations and we can't Perform Addition of New Objects and Replacing Existing Objects.

To Overcome these Limitations we should go for ListIterator.

ListIterator

Overview

- ListIterator is the Child Interface of Iterator
- By using ListIterator we can Move Either to the Forward Direction OR to the Backward Direction. That is it is a Bi-Directional Cursor
- Able to read and remove object
- Also able to Perform Addition of New Objects and Replacing existing Objects
- We can Create a ListIterator object by using listIterator().
  public ListIterator listIterator();
- eg.,
  List list = new ArrayList<>();
  ListIterator Itr = list.listIterator();



Methods

- ListIterator is the Child Interface of Iterator and Hence All Iterator Methods by Default available to the ListIterator.

- ListIterator Defines the following 9 Methods.
  - public boolean hasNext()
  - public Object next()
  - public int nextIndex()
  - public boolean hasPrevious()
  - public Object previous()
  - public int previousIndex()
  - public void remove()
  - public void set(Object new)
  - public void add(Object new)

Limitation

- The Most Powerful Cursor is ListIterator. But its Limitation is, it is Applicable Only for List Objects.

Comparison of 3 cursors

Enumeration	**Iterator**	**ListIterator**
Applicable for Legacy classes	Applicable for Collection	Applicable for only List Objects
Only Forward Direction	Only Forward Direction	Bi-Direction
Get element by using element()	Get element by using iterator()	Get element by using listIterator() of List (I)
Perform only Read Operation	Perform Read and remove operation	Perform Read, Remove, Replace, Add new element Operations

hasMoreElements()
nextElement()
	hasNext()
	next()
	remove()
		hasNext()
		next()
		nextIndex()
		hasPrevious()
		previous()
		previousIndex()
		remove()
		set(Object new)
		add(Object new)

introduced in 1.0 V	introduced in 1.2 V	introduced in 1.2 V