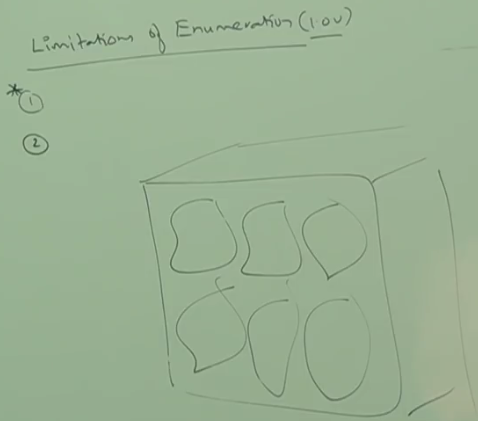
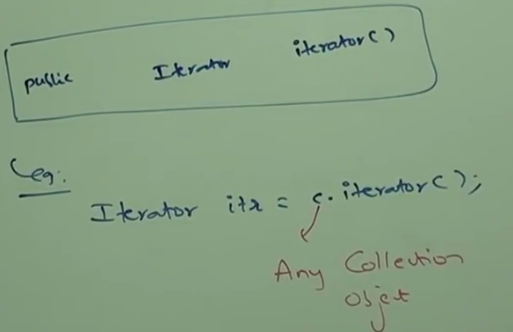
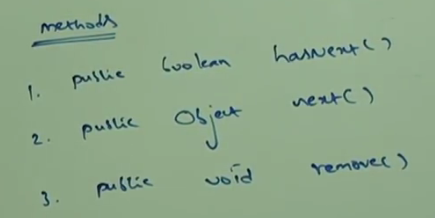
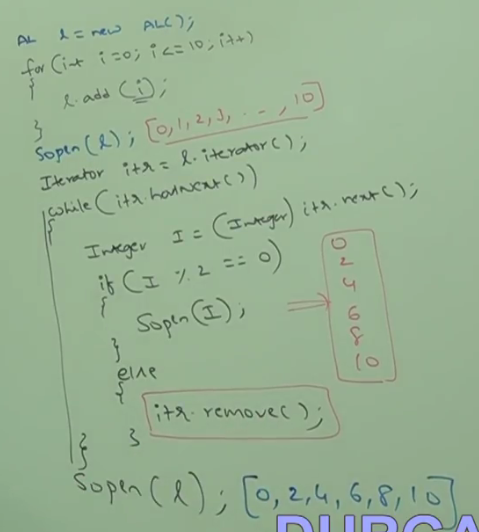
# Limitations with Enumeration

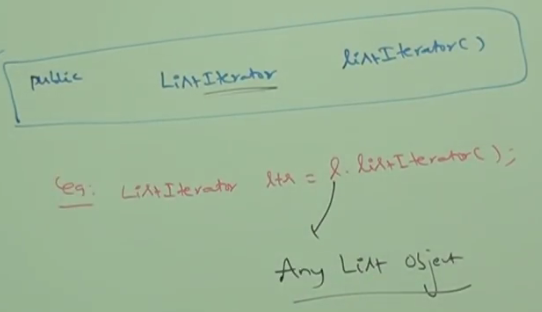
**NOTE**: Enumeration came in 1.0.

1. We can apply Enumeration concept only for legacy classes and it’s not a universal cursor.
2. By using Enumeration, we can get only read-access and we can’t perform **remove operation**.   
     
   To overcome above limitations, we should go for **Iterator**
3. **d**

Iterator Cursor

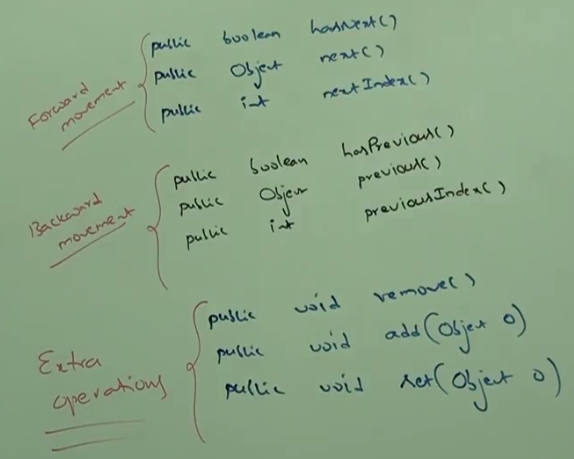
1. **Universal** Cursor:We can apply Iterator Concept for any collection and hence it’s **universal cursor**.
2. **Read & Write**: By using Iterator, we can perform both **read and remove operations**.  
   So Iterator solves the limitations found in Enumeration.
3. **How to get Iterator**
   1. ****
4. **Methods in Iterator:  
   **
5. **Demonstrating the remove operation in Iterator  
   **
6. **Limitations of Iterator:**
   1. **One-Directional Cursor**: By using Enumeration and Iterator, we can always move only forward not backward direction. These are single-direction cursor. But not bi-directional cursor.
   2. **Replace & Add**: Using Iterator and Enumeration, We can read and remove operations and we can’t perform replacement and addition of new object.
   3. **NOTE:** To overcome above limitations, we should go for ListIterator.
7. **d**

ListIterator

1. By using ListIterator, we can move in both forward and backward directions. Hence, it’s **bi-directional cursor**.
2. By using ListIterator, we can perform replacement and addition of new object in addition to “read and remove” operations.
3. **How to get ListIterator**
   1. 
4. ListIterator is child Interface of Iterator and hence, all methods in Iterator are by default available to ListIterator.

ListIterator

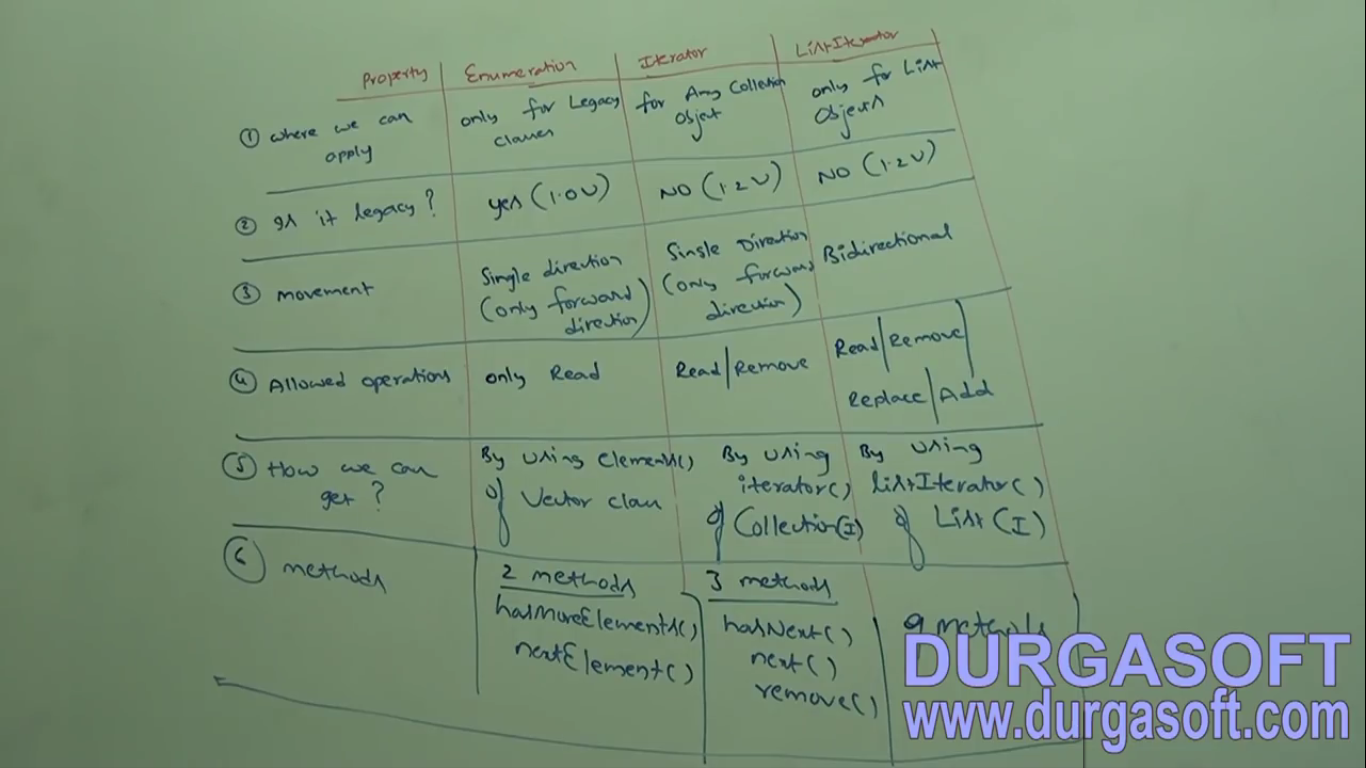
Iterator

1. **Methods in ListIterator:** 12 Methods
   1. 
2. d

# Limitation of ListIterator

1. The most powerful cursor is ListIterator but its limitation is, it’s applicable only for List object.

Comparison Table of 3 Cursors



Enumeration, Iterator and ListIterator are interfaces their implemented classes:

