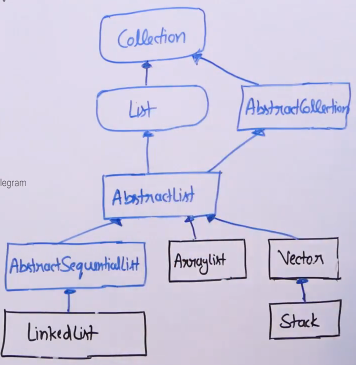
**List Interface in Java**



List Opertions

* get(int index)
* set(int index, element)
* indexOf(object o)
* listItrator()
* listItrator(int index)
* remove(int index)

**Syntax for declaring a list:**

List<Integer>list = new Arraylist<Integer>()

**ListIterator in Java**

* Works only for list (Linkedlist, Arraylist etc)
* Inherites from Iterator and provides below additional functionalities (in addition to next(), hasNext() and remove().
  + hasPrevious()
  + previous()
  + add()
  + set()
  + nextIndex()
  + previousIndex()

Methods of Iterator

* next()
* hasNext()
* remove()

**Forward Traversal in List**

```

Class GFG{

public static void main(String args[]) {

list.add(10);

list.add(20);

list.add(30);

ListIterator<Integer> it = list.listIterator();

While(it.hasNext(1)){

System.out.println(it.next(1));

}

}

}

```

**Backward Traversal**

```

Class GFG{

public static void main(String args[]) {

list.add(10);

list.add(20);

list.add(30);

ListIterator<Integer> it = list.listIterator(list.size());

While(it.hasPrevious(1)){

System.out.println(it.previous(1));

}

}

}

```

**Set Method**

```

Class GFG{

public static void main(String args[]) {

list.add(10);

list.add(20);

list.add(30);

ListIterator<Integer> it = list.listIterator(list.size());

While(it.hasPrevious(1)){

int x = (Integer)it.previous();

it.set(s\*2);

}

System.out.println(list);

}

}  
```

Output: [20, 40, 60]

**Add Method**

```

Class GFG{

public static void main(String args[]) {

list.add(10);

list.add(20);

list.add(30);

ListIterator<Integer> it = list.listIterator(list.size());

While(it.hasNext(1)){

It.add(5);

it.next();

}

System.out.println(list);

}

}

```

Output: [5,10,5,20,5,30,5]

**nextIndex & previousIndex()**

```

Class GFG{

public static void main(String args[]) {

list.add(10);

list.add(20);

list.add(30);

ListIterator<Integer> it = list.listIterator(list.size(2));

System.out.println(it.previousIndex());

System.out.println(it.nextIndex());

```

Output: 2