Java Collections Part 6(ArrayList iteration Methods)

There are few ways to iterate over the elements of ArrayList<E>.

|  |  |
| --- | --- |
| Method | What it uses? |
| Using random access to array | size() and get(index) |
| For-each loop | Iterate over array or Iterable<E> |
| Using iterator() of List<E> | hasNext() and next() of Iterator<E> |
| Using listIterator() of List<E> | hasNext(), nextIndex() and next() of ListIterator<E> |
| Using listIterator() of List<E> | hasPrevious(), previousIndex() and previous() of ListIterator<E> |
| forEach loop | Lambda expression, print as action. |

Below problem walk you through entire details of the iteration over ArrayList<E>

**package** arraylist;

**import** java.util.ArrayList;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.ListIterator;

**public** **class** ArrayListTraversal {

**public** **static** **void** main(String[] args) {

/\*\*

\* Populate list with some integer values

\* \*/

List<Integer> list = **new** ArrayList<Integer>();

list.add(1);

list.add(78);

list.add(23);

list.add(19);

list.add(7);

list.add(10);

/\*\*

\* As we have discussed List<E> is backed by an array and

\* hence it provides us RandomAccess to elements.

\* We leverage the concept of E get(int index); method

\* from List<E> interface and traverse till the end of List<E>.

\* \*/

System.***out***.print("Iteration using Random Access to Array index ");

**for** (**int** i = 0; i < list.size(); i++) {

System.***out***.print(list.get(i) + " ");

}

System.***out***.println();

System.***out***.println();

/\*\*

\* This is foreach loop.

\* It is used to iterate over array or Iterable<E>

\* Pronounce it as for each number of type Integer in list print it.

\* \*/

System.***out***.print("Using for-each loop ");

**for** (Integer number : list) {

System.***out***.print(number + " ");

}

/\*\*

\* Use iterator to iterate through the list

\* hasNext()

\* return true if list has next element.

\* return false if end of list is reached.

\* next()

\* returns the next element in list.

\* \*/

System.***out***.print("Using Itertor ");

Iterator<Integer> iterator = list.iterator();

**while** (iterator.hasNext()) {

System.***out***.print(iterator.next() + " ");

}

System.***out***.println();

System.***out***.println();

/\*\*

\* listIterator is used to traverse list is front and back direction.

\* hasNext()

\* return true if list has next element.

\* return false if end of list is reached.

\* next()

\* return the next element in list.

\* \*/

System.***out***.print("Using list Iterator moving forward ");

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

**while** (listItr.hasNext()) {

System.***out***.print(listItr.nextIndex() + "-" + listItr.next() + " ");

}

System.***out***.println();

System.***out***.println();

/\*\*

\* listiterator's hasPrevious(), previousIndex() and previous() is used.

\* hasPrevious()

\* return true if list has previous element.

\* return false if list has reached start of list.

\*

\* previousIndex()

\* returns index of previous element without moving the pointer to previous location.

\*

\* previous()

\* returns previous element in list.

\* \*/

System.***out***.print("Using list Iterator moving backward ");

**while** (listItr.hasPrevious()) {

System.***out***.print(listItr.previousIndex() + "-" + listItr.previous()+ " ");

}

System.***out***.println();

System.***out***.println();

/\*\*

\* Below method uses lambda expression to traverse in list.

\* This method will work only in Java 8.

\* forEach method performs the given action on each element in list.

\* Here action to be performed is printing every element of list.

\* \*/

System.***out***.print("Using foreach loop and lambda ");

list.forEach(number -> System.***out***.print(number+" "));

}

}