TECHPROED

PROFESSIONAL TECHNOLOGY EDUCATION

# WELCOME TO TECHPROED JAVA TUTORIAL

**Testi baslatmak icin asagidaki adımları takip ediniz**

**Go to** www.socrative.com

***Click on Login***

***Click on Student Login***

***Room Name*: A L P T E K I N 3 5 2 3**

***Kayıtta kullandığınız ismi tam olarak yazınız***

***Time: 11 Minutes***

TECHPROED
PROFESSIONAL TECHNOLOGY EDUCATION

# Iterators

## 1) Iterator

```java
package day29_collections;

import java.util.Iterator;

public class IteratorAndForEachLoop {

    public static void main(String[] args){

        List<String> list2 = new LinkedList<String>();
        list2.add("X");
        list2.add("Y");
        list2.add("Z");
        System.out.println(list2);

        //Iterator is used to modify the collections.
        //It has three methods
        //       1)hasNext():Returns true if there are more elements.Otherwise,returns false.
        //       2)next():Returns the next element.Throws NoSuchElementException if there is not a next element.
        //       3)remove():Removes the current element.
        //         Throws IllegalStateException if you call remove( ) before using next( ).

        Iterator<String> iterator1 = list2.iterator();
        //To print elements we can use next() method
        while(iterator1.hasNext()) {
            System.out.println(iterator1.next());
        }

        //We can use for-each() loop to print elements on the console as well
        for(String w:list2) {
            System.out.println(w);
        }

        //What is the difference between for-each() and iterator() ?
        //We cannot modify a collection by using for-each(); however, iterator() can modify.
        for(String w:list2) {
            w=w+"M";
        }
        System.out.println("After for-each() loop: " + list2);

        //Before every while you need to use iterator()
        Iterator<String> iterator2 = list2.iterator();
        while(iterator2.hasNext()) {
            iterator2.next(); // Do not forget to use next() before using remove()
            iterator2.remove();
        }
        System.out.println("After iterator(): " + list2);
    }
}
```

# 2) ListIterator

```java
package day29_collections;

import java.util.LinkedList;

public class ListIteratorMethods01 {

	public static void main(String[] args) {

	  List<String> list = new LinkedList<String>();
	  list.add("A");
	  list.add("B");
	  list.add("C");
	  System.out.println("List: " + list);

	  System.out.println("List is in the given order");

	  //To print elements on the console
	  ListIterator<String> listItr1 = list.listIterator();
	  while(listItr1.hasNext()){
	      Object element = listItr1.next();
	      System.out.println(element);
	  }

	  System.out.println("List is reversed");

	  //To display the list backwards use hasPrevious() and previous()
	  //Before using hasPrevious() and previous(), you need to use hasNext() and next()
	  //to move the pointer to the end of the list
	  while(listItr1.hasPrevious()){
	      Object element = listItr1.previous();
	      System.out.println(element);
	  }

	  System.out.println("List is updated");

	  //To update the elements use set()
	  ListIterator<String> listItr2 = list.listIterator();
	  while(listItr2.hasNext()){
	      Object element = listItr2.next();
	      listItr2.set(element + "W");
	  }
	  System.out.println("Updated list: " + list);

	}
}
```

```java
package day29_collections;

import java.util.LinkedList;
import java.util.List;
import java.util.ListIterator;

public class ListIteratorMethods02 {

    public static void main(String[] args) {

        List<String> list = new LinkedList<String>();
        list.add("A");
        list.add("B");
        list.add("C");
        System.out.println("List: " + list);

        //To remove all elements use remove() after next();
        //if you do not use next you will get IllegalStateException
        ListIterator<String> listItr1 = list.listIterator();
        while(listItr1.hasNext()){
            listItr1.next();
            listItr1.remove();
        }
        System.out.println("After removing: " + list);

        //To add new elements to the list use add()
        listItr1.add("X");
        listItr1.add("Y");
        listItr1.add("Z");
        System.out.println("After adding: " + list);


    }

}
```