

# List的介绍

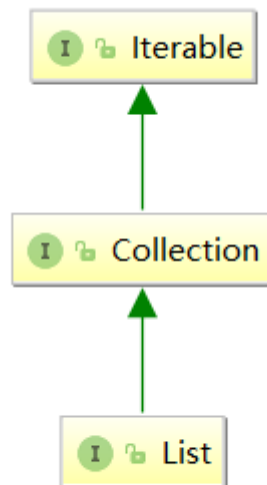
---

## 【本节目标】

1. 什么是List
2. List常见接口介绍
3. List的使用

## 1. 什么是List

在集合框架中，List是一个接口，继承自Collection。



Collection也是一个接口，该接口中规范了后序容器中常用的一些方法，具体如下所示：

I Collection		
m	size()	int
m	contains(Object)	boolean
m	iterator()	Iterator<E>
m	toArray()	Object[]
m	toArray(T[])	T[]
m	add(E)	boolean
m	remove(Object)	boolean
m	containsAll(Collection<?>)	boolean
m	addAll(Collection<? extends E>)	boolean
m	removeAll(Collection<?>)	boolean
m	removeIf(Predicate<? super E>)	boolean
m	retainAll(Collection<?>)	boolean
m	clear()	void
m	equals(Object)	boolean
m	hashCode()	int
m	splitterator()	Splitterator<E>
m	stream()	Stream<E>
m	parallelStream()	Stream<E>
p	empty	boolean

Iterable也是一个接口，表示实现该接口的类是可以逐个元素进行遍历的，具体如下：

I Iterable		
m	iterator()	Iterator<T>
m	forEach(Consumer<? super T>)	void
m	splitterator()	Splitterator<T>

[List 的官方文档](#)

站在数据结构的角度来看，**List**就是一个线性表，即n个具有相同类型元素的有限序列，在该序列上可以执行增删改查以及变量等操作。

【面试题】Collection中有那些方法？

## 2. 常见接口介绍

List中提供了好的方法，具体如下：

I List		
m	size()	int
m	contains(Object)	boolean
m	iterator()	Iterator<E>
m	toArray()	Object[]
m	toArray(T[])	T[]
m	add(E)	boolean
m	remove(Object)	boolean
m	containsAll(Collection<?>)	boolean
m	addAll(Collection<? extends E>)	boolean
m	addAll(int, Collection<? extends E>)	boolean
m	removeAll(Collection<?>)	boolean
m	retainAll(Collection<?>)	boolean
m	replaceAll(UnaryOperator<E>)	void
m	sort(Comparator<? super E>)	void
m	clear()	void
m	equals(Object)	boolean
m	hashCode()	int
m	get(int)	E
m	set(int, E)	E
m	add(int, E)	void
m	remove(int)	E
m	indexOf(Object)	int
m	lastIndexOf(Object)	int
m	listIterator()	ListIterator<E>
m	listIterator(int)	ListIterator<E>
m	subList(int, int)	List<E>
m	splitIterator()	SplitIterator<E>
p	empty	boolean

Generated by Vfiles

虽然方法比较多，但是常用方法如下：

方法	解释
boolean <a href="#">add</a> (E e)	尾插 e
void <a href="#">add</a> (int index, E element)	将 e 插入到 index 位置
boolean <a href="#">addAll</a> (Collection<? extends E> c)	尾插 c 中的元素
E <a href="#">remove</a> (int index)	删除 index 位置元素
boolean <a href="#">remove</a> (Object o)	删除遇到的第一个 o
E <a href="#">get</a> (int index)	获取下标 index 位置元素
E <a href="#">set</a> (int index, E element)	将下标 index 位置元素设置为 element
void <a href="#">clear</a> ()	清空
boolean <a href="#">contains</a> (Object o)	判断 o 是否在线性表中
int <a href="#">indexOf</a> (Object o)	返回第一个 o 所在下标
int <a href="#">lastIndexOf</a> (Object o)	返回最后一个 o 的下标
List<E> <a href="#">subList</a> (int fromIndex, int toIndex)	截取部分 list

### 3. List的使用

注意：**List是个接口，并不能直接用来实例化。**

如果要使用，必须去实例化List的实现类。在集合框架中，**ArrayList和LinkedList都实现了List接口。**

具体使用参考下个课件。