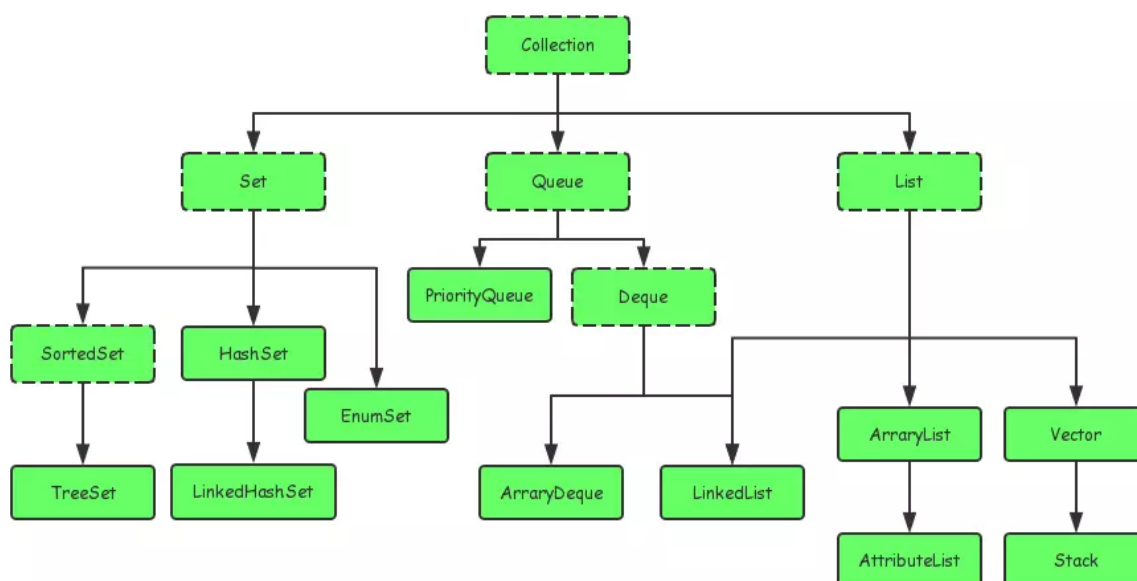


Collection接口是Set, Queue, List的父接口。Collection接口中定义了多种方法可供其子类进行实现，以实现数据操作。



1. Set

- **TreeSet**：基于红黑树实现，支持有序性操作，例如根据一个范围查找元素的操作。但是查找效率不如 HashSet，HashSet 查找的时间复杂度为 $O(1)$ ，TreeSet 则为 $O(\log N)$ 。
- **HashSet**：基于哈希表实现，支持快速查找，但不支持有序性操作。并且失去了元素的插入顺序信息，也就是说使用 Iterator 遍历 HashSet 得到的结果是不确定的。
- **LinkedHashSet**：具有 HashSet 的查找效率，且内部使用双向链表维护元素的插入顺序。

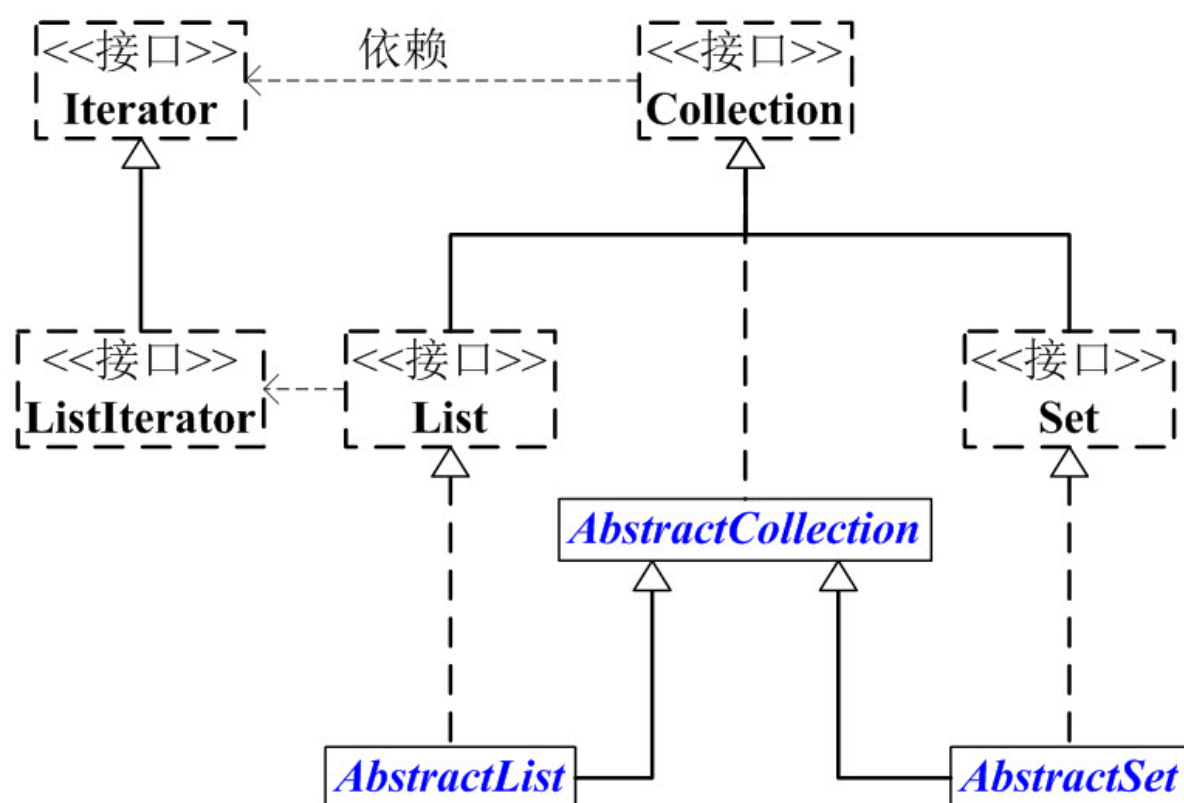
2. List

- **ArrayList**：基于动态数组实现，支持随机访问。
- **Vector**：和 ArrayList 类似，但它是线程安全的。
- **LinkedList**：基于双向链表实现，只能顺序访问，但是可以快速地在链表中间插入和删除元素。不仅如此，LinkedList 还可以

用作栈、队列和双向队列。

3. Queue

- LinkedList：可以用它来实现双向队列。
- PriorityQueue：基于堆结构实现，可以用它来实现优先队列。



Collection接口

Collection接口的所有子类必须实现2种构造函数：不带参数的构造函数 和 参数为Collection的构造函数。带参数的构造函数，可以用来转换Collection的类型。

方法摘要	
boolean	<code>add(E e)</code> 确保此 collection 包含指定的元素（可选操作）。
boolean	<code>addAll(Collection<? extends E> c)</code> 将指定 collection 中的所有元素都添加到此 collection 中（可选操作）。
void	<code>clear()</code> 移除此 collection 中的所有元素（可选操作）。
boolean	<code>contains(Object o)</code> 如果此 collection 包含指定的元素，则返回 true。
boolean	<code>containsAll(Collection<?> c)</code> 如果此 collection 包含指定 collection 中的所有元素，则返回 true。
boolean	<code>equals(Object o)</code> 比较此 collection 与指定对象是否相等。
int	<code>hashCode()</code> 返回此 collection 的哈希码值。
boolean	<code>isEmpty()</code> 如果此 collection 不包含元素，则返回 true。
Iterator<E>	<code>iterator()</code> 返回在此 collection 的元素上进行迭代的迭代器。
boolean	<code>remove(Object o)</code> 从此 collection 中移除指定元素的单个实例，如果存在的话（可选操作）。
boolean	<code>removeAll(Collection<?> c)</code> 移除此 collection 中那些也包含在指定 collection 中的所有元素（可选操作）。
boolean	<code>retainAll(Collection<?> c)</code> 仅保留此 collection 中那些也包含在指定 collection 的元素（可选操作）。
int	<code>size()</code> 返回此 collection 中的元素数。
Object[]	<code>toArray()</code> 返回包含此 collection 中所有元素的数组。
<T> T[]	<code>toArray(T[] a)</code> 返回包含此 collection 中所有元素的数组；返回数组的运行时类型与指定数组的运行时类型相同。

AbstractCollection抽象类

实现了Collection中除iterator()和size()之外的函数。方便其它类实现Collection，比如ArrayList、LinkedList等

AbstractList抽象类

实现List接口除size()、get(int location)之外的函数的抽象类。和AbstractCollection相比，AbstractList抽象类中，实现了iterator()接口。

提供了相关的添加、删除、修改、遍历等功能。

AbstractSet抽象类

继承于AbstractCollection，并且实现Set接口的抽象类。和AbstractCollection一样。

Iterator接口

集合的迭代器。Iterator遍历Collection时，是fail-fast机制的。

ListIterator接口

继承于Iterator的接口，是队列迭代器。相比于Iterator，它新增了添加、是否存在上一个元素、获取上一个元素等等API接口。

1. java 中的length 属性是针对数组说的,比如说你声明了一个数组,想知道这个数组的长度则用到了 length 这个属性.
2. java 中的length()方法是针对字符串String说的,如果想看这个字符串的长度则用到 length()这个方法.
3. .java 中的size()方法是针对泛型集合说的,如果想看这个泛型有多少个元素,就调用此方法来查看!