Java核心API (集合)

# Collection集合框架

|  |
| --- |
| Java.util.Collection接口  |--List子接口  |--ArrayList实现类  |--LinkedList实现类  |--Vector实现类  |--Stack(栈，继承Vector，先进后出)  |--Set子接口  |--HashSet实现类  |--TreeSet实现类  |--Queue接口(普通队列，先进先出)  |--Deque子接口(双端队列)  |--ArrayDeque实现类 |
| 集合泛型支持：  1）泛型是 JDK1.5引入的新特性，泛型的本质是参数化类型。在类、接口、方法的定义过程中，所操作的数据类型为传入的指定参数类型。所有的集合类型都带有泛型参数，这样在创建集合时可以指定放入集合中的对象类型。同时，编译器会以此类型进行检查。  2）ArrayList支持泛型，泛型尖括号里的符号可随便些，但通常大写E。  3）迭代器支持泛型，但是迭代器使用的泛型应该和它所迭代集合的泛型类型一致！  4）泛型只支持引用类型，不支持基本类型，但可以使用对应的包装类  5）如果泛型不指定类型的话，默认为Object类型。 |

## List接口和Set接口

List接口：List接口是Collection的子接口，用于定义线性表数据结构，元素可重复、有序的；可以将List理解为存放对象的数组，只不过其元素个数可以动态的增加或减少。

(1) List接口的两个常见的实现类：ArrayList和LinkedList，分别用动态数组和链表的方式实现了List接口。List、ArrayList和LinkedList均处于java.util包下。

(2) 可以认为ArrayList和LinkedList的方法在逻辑上完全一样，只是在性能上有一定的差别，ArrayList更适合于随机访问，而LinkedList更适合于插入和删除，在性能要求不是特别苛刻的情形下可以忽略这个差别。

(3) 使用List我们不需要在创建的时候考虑容量集合的容量是根据其所保存的元素决定的换句话说，集合的容量是可以自动扩充的。

(4) List的实现类会重写toString方法，依次调用所包含对象的toString方法，返回集合中所包含对象的字符串表现 。

2）Set接口：Set接口是Collection的子接口，用于存储不重复的元素，元素是无序的。

(1) HashSet和TreeSet是Set集合的两个常见的实现类，分别用hash表和排序二叉树的方式实现了Set集合。HashSet是使用散列算法实现Set的。

(2) Set集合没有get(int index)方法，我们不能像使用List那样，根据下标获取元素。想获取元素需要使用Iterator。

(3) 向集合添加元素也使用add方法，但是add方法不是向集合末尾追加元素，因为无序。