



当前位置: HollisChuang's Blog (<https://www.hollischuang.com>) > Java (<https://www.hollischuang.com/archives/category/java>) > 正文

ArrayList vs LinkedList vs Vector 区别 (<https://www.hollischuang.com/archives/1349>)

2016-03-23 分类: Java (<https://www.hollischuang.com/archives/category/java>) 阅读(11232) 评论(0)

[GitHub 19k Star 的Java工程师成神之路，不来了解一下吗！](https://github.com/hollischuang/toBeTopJavaer)
(<https://github.com/hollischuang/toBeTopJavaer>)

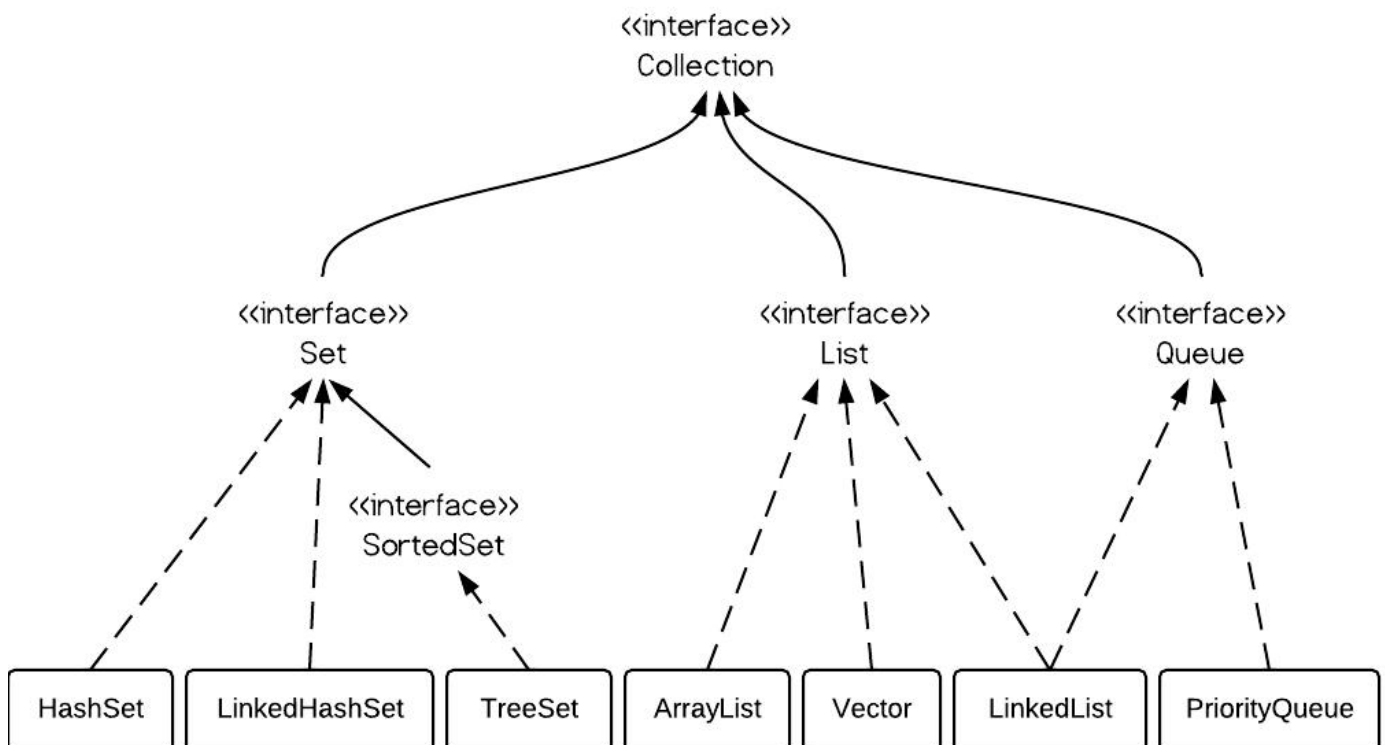
List

List 是一个接口，它继承于Collection的接口。它代表着有序的队列。当我们讨论List的时候，一般都和Set作比较。

List中元素可以重复，并且是有序的（这里的有序指的是按照放入的顺序进行存储。如按照顺序把1, 2, 3存入List，那么，从List中遍历出来的顺序也是1, 2, 3）。

Set中的元素不可以重复，并且是无序的（从set中遍历出来的数据和放入顺序没有关系）。

下面是Java中的集合类的关系图。从中可以大致了解集合类之间的关系



(<http://www.hollischuang.com/wp-content/uploads/2016/03/java-collection-hierarchy.jpeg>)

ArrayList、LinkedList 和 Vector之间的区别

从上图可以看出，ArrayList、LinkedList 和 Vector都实现了List接口，是List的三种实现，所以在用法上非常相似。他们之间的主要区别体现在不同操作的性能上。后面会详细分析。

ArrayList

ArrayList底层是用数组实现的，可以认为ArrayList是一个可改变大小的数组。随着越来越多的元素被添加到ArrayList中，其规模是动态增加的。

LinkedList

LinkedList底层是通过双向链表实现的。所以，LinkedList和ArrayList之前的区别主要就是数组和链表的差别。

数组中查询和赋值比较快，因为可以直接通过数组下标访问指定位置。

链表中删除和增加比较快，因为可以直接通过修改链表的指针（Java中并无指针，这里可以简单理解为指针。其实是通过Node节点中的变量指定）进行元素的增删。

所以，LinkedList和ArrayList相比，增删的速度较快。但是查询和修改值的速度较慢。同时，LinkedList还实现了Queue接口，所以他还提供了offer(), peek(), poll()等方法。

Vector


Vector和ArrayList一样，都是通过数组实现的，但是Vector是线程安全的。和ArrayList相比，其中的很多方法都通过同步（synchronized）处理来保证线程安全。

如果你的程序不涉及到线程安全问题，那么使用ArrayList是更好的选择（因为Vector使用synchronized，必然会影响效率）。

二者之间还有一个区别，就是扩容策略不一样。在List被第一次创建的时候，会有一个初始大小，随着不断向List中增加元素，当List认为容量不够的时候就会进行扩容。Vector缺省情况下自动增长原来一倍的数组长度，ArrayList增长原来的50%。

ArrayList 和 LinkedList的性能对比

使用以下代码对ArrayList和LinkedList中的几种主要操作所用时间进行对比：



```
// ArrayList add
long startTime = System.nanoTime();

for (int i = 0; i < 100000; i++) {
    arrayList.add(i);
}
long endTime = System.nanoTime();
long duration = endTime - startTime;
System.out.println("ArrayList add: " + duration);

// LinkedList add
startTime = System.nanoTime();

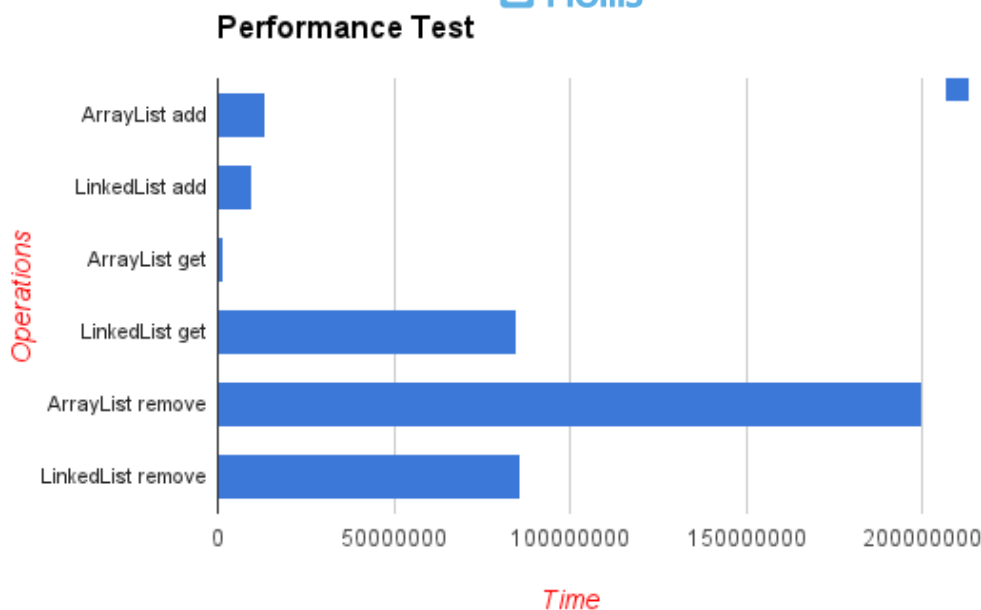
for (int i = 0; i < 100000; i++) {
    linkedList.add(i);
}
endTime = System.nanoTime();
duration = endTime - startTime;
System.out.println("LinkedList add: " + duration);

// ArrayList get
startTime = System.nanoTime();

for (int i = 0; i < 10000; i++) {
    arrayList.get(i);
}
endTime = System.nanoTime();
duration = endTime - startTime;
System.out.println("ArrayList get: " + duration);
```

结果:

```
ArrayList add: 13265642
LinkedList add: 9550057
ArrayList get: 1543352
LinkedList get: 85085551
ArrayList remove: 199961301
LinkedList remove: 85768810
```



(<http://www.hollischuang.com/wp-content/uploads/2016/03/arraylist-vs-linkedlist1.png>).

他们的表现的差异是显而易见的。在添加和删除操作上LinkedList更快,但在查询速度较慢。

如何选择

如果涉及到多线程，那么就选择Vector（当然，你也可以使用ArrayList并自己实现同步），如果不涉及到多线程就从LinkedList、ArrayList中选。LinkedList更适合从中间插入或者删除（链表的特性）。ArrayList更适合检索和在末尾插入或删除（数组的特性）。

参考资料

[ArrayList vs. LinkedList vs. Vector](http://www.programcreek.com/2013/03/arraylist-vs-linkedlist-vs-vector/) (<http://www.programcreek.com/2013/03/arraylist-vs-linkedlist-vs-vector/>).

(全文完)

扫描二维码，关注作者微信公众号