# 浅谈java集合中线程安全的类

vector：就比arraylist多了个同步化机制（线程安全），因为效率较低，现在已经不太建议使用。在web应用中，特别是前台页面，往往效率（页面响应速度）是优先考虑的。

statck：堆栈类，先进后出

hashtable：就比hashmap多了个线程安全

Collections的synchronizedXxxx（）方法包装的集合

ConcurrentXxxx：从jdk1.5提供，通过分段锁实现线程安全，使用这类方法既可以不怎么影响效率，又可以保证安全，建议使用。

除了这些之外，其他的都是非线程安全的类和接口。

线程安全的类其方法是同步的，每次只能一个访问。是重量级对象，效率较低，一般用的不多，只有在特殊情况下会考虑使用。

1、Vector、ArrayList、LinkList之间的区别

Vector : 基于Array的List，其实就是封装了Array所不具备的一些功能方便我们使用，它不可能走出Array的限制。性能也就不可能超越Array。所以，在可能的情况下，我们要多运用Array。另外很重要的一点就是Vector“synchronized”的，这个也是Vector和ArrayList的唯一的区别。

[ArrayList](http://baike.baidu.com/item/ArrayList" \t "http://baike.baidu.com/_blank)：同Vector一样是一个基于数组实现的，但是不同的是ArrayList不是同步的。所以在性能上要比Vector优越一些，但是当运行到多线程环境中时，可需要自己在管理线程的同步问题。

LinkedList：LinkedList不同于前面两种List，它不是基于[Array](http://baike.baidu.com/item/Array" \t "http://baike.baidu.com/_blank)的，所以不受Array性能的限制。它每一个节点（Node）都包含两方面的内容：1.节点本身的数据（data）；2.下一个节点的信息（nextNode）。所以当对LinkedList做添加，删除动作的时候就不用像基于Array的List一样，必须进行大量的数据移动。只要更改nextNode的相关信息就可以实现了。这就是LinkedList的优势。

2、HashTable跟HashMap的区别

HashTable是线程安全的,即HashTable的方法都提供了同步机制；HashMap不是线程安全的,即不提供同步机制 ；HashTable不允许插入空值,HashMap允许!

3、StringBuffer和StringBuilder的区别

StringBuffer是线程安全的，StringBuilder是线程不安全的。