

java认证:Java中静态数组与动态数组Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/582/2021_2022_java_E8_AE_A4_E8_AF_81_c104_582933.htm 我们学习的数组都是静态数组，其实在很多的时候，静态数组根本不能满足我们编程的实际需要，比方说我需要程序运行过程中动态的向数组中添加数据，这时我们的静态数组大小是固定的，显然就不能添加数据，要动态添加数据必须要用到动态数组，动态数组中的各个元素类型也是一致的，不过这种类型已经是用一个非常大的类型来揽括Object类型。Object类是JAVA.LANG包中的顶层超类。所有的类型都可以与Object类型兼容，所以我们可以将任何Object类型添加至属于Object类型的数组中，能添加Object类型的集合有ArrayList、Vector及LinkedList，它们对数据的存放形式仿造于数组，属于集合类，下面是他们的特点：
特点一、容量扩充性 从内部实现机制来讲ArrayList和Vector都是使用Object的数组形式来存储的。当你向这两种类型中增加元素的时候，如果元素的数目超出了内部数组目前的长度它们都需要扩展内部数组的长度，Vector缺省情况下自动增长原来一倍的数组长度，ArrayList是原来的50%，所以最后你获得的这个集合所占的空间总是比你实际需要的要大。所以如果你要在集合中保存大量的数据那么使用Vector有一些优势，因为你可以通过设置集合的初始化大小来避免不必要的资源开销。
特点二、同步性 ArrayList,LinkedList是不同步的，而Vector是的。所以如果要求线程安全的话，可以使用ArrayList或LinkedList，可以节省为同步而耗费开销。但在多线程的情况下，有时候就不得不使用Vector了。当然，也

可以通过一些办法包装ArrayList,LinkedList，使他们也达到同步，但效率可能会有所降低。

特点三、数据操作效率

ArrayList和Vector中，从指定的位置(用index)检索一个对象，或在集合的末尾插入、删除一个对象的时间是一样的，可表示为 $O(1)$ 。但是，如果在集合的其他位置增加或移除元素那么花费的时间会呈线性增长： $O(n-i)$ ，其中 n 代表集合中元素的个数， i 代表元素增加或移除元素的索引位置。为什么会这样呢?以为在进行上述操作的时候集合中第 i 和第 i 个元素之后的所有元素都要执行 $(n-i)$ 个对象的位移操作。LinkedList中，在插入、删除集合中任何位置的元素所花费的时间都是一样的 $O(1)$ ，但它在索引一个元素的时候比较慢，为 $O(i)$ ，其中 i 是索引的位置。所以，如果只是查找特定位置的元素或只在集合的末端增加、移除元素，那么使用Vector或ArrayList都可以。如果是对其它指定位置的插入、删除操作，最好选择LinkedList

ArrayList和Vector是采用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，都允许直接序号索引元素，但是插入数据要设计到数组元素移动等内存操作，所以索引数据快插入数据慢，Vector由于使用了synchronized方法(线程安全)所以性能上比ArrayList要差，LinkedList使用双向链表实现存储，按序号索引数据需要进行向前或向后遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

更多优质资料尽在百考试题论坛 百考试题在线题库 java认证更多详细资料 100Test 下载频道 开通，各类考试题目直接下载。详细请访问 www.100test.com