数组是在物理内存上一片连续的、容量确定的存储空间。如int[10]。

基本数据类型中的数组，不能动态增加、删除节点（元素），所以产生了线性表。

线性表包含：

ArrayList特点：

**物理上连续**

**逻辑上连续**

大小可**动态增加**的数组

ArrayList添加或删除节点，要涉及到大多数节点的位移，耗费内存和时间非常多，造成代码性能降低，所以它的增删慢，但是可以使用链表来解决这个问题。

LinkedList是链表，他的特点：

**物理上不连续**

**逻辑上连续**

可以动态增加和删除节点

LinkedList添加或删除节点，只要改变相关节点的link.previous或者link.next值，只是赋值过程，所以添删节点效率高。而它的查询需要轮寻查找，所以它的查询慢。

一句话都是因为他是链表特点，物理不连续，逻辑连续的造成的。

有没有结合两者的优点（查询、增删都快）的数据结构？

有，Hash表。

Hash算法：

Hash%length=0~15 === hash & （length-1）

Hash冲突