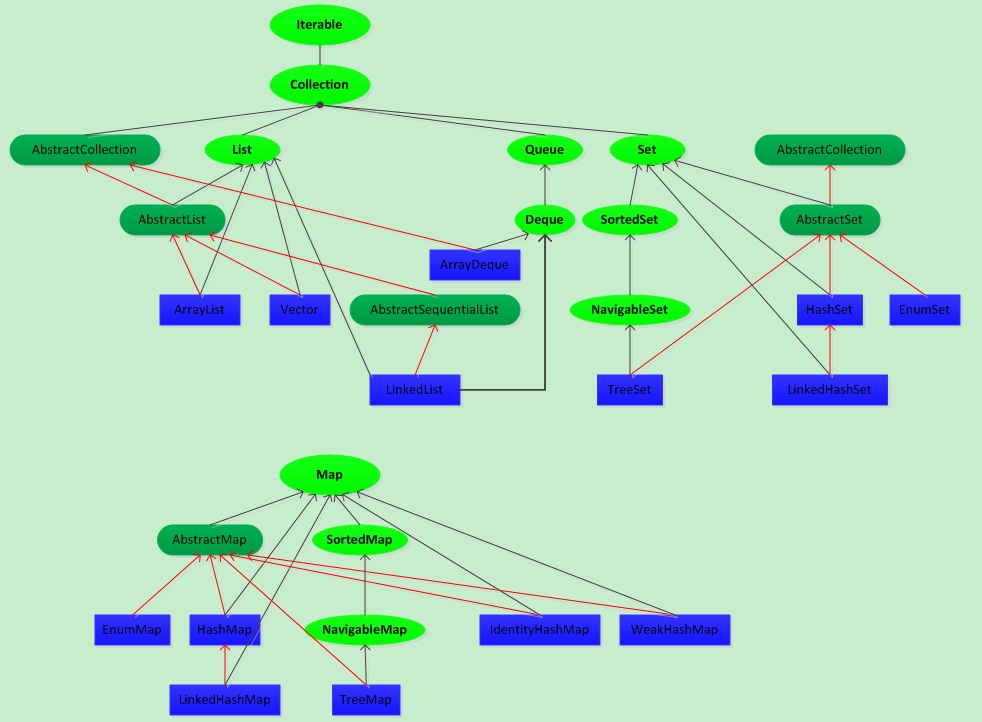
1. java集合

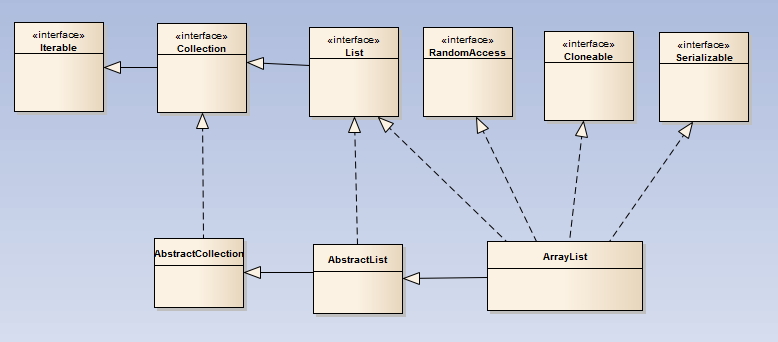
java集合主要包含list，set，map三大类，list的特点是有序可重复，set的特点是无序不可重复，map是存储key-value值。List的实现主要有**ArrayList**，Vector和**LinkedList**;

set的实现有**HashSet**,**TreeSet**,EnumSet,LinkedHashSet; map的实现有HashMap,TreeMap,EnumMap, LinkedHashMap。

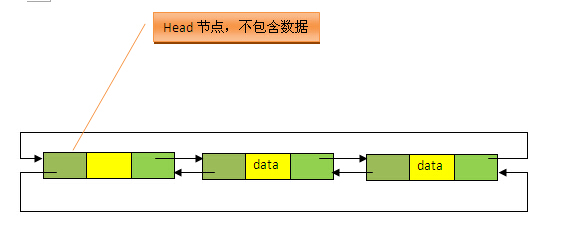


1. list

**ArrayList**：内部实现方式是基于数组进行实现的，ArrayList初始化为{}，初始化长度为10，当需要扩容时，新长度为原来的1.5倍，扩容时不需要添。常用的方法有add(),addAll(),contains(),remove(),removeAll(),get(),set()等方法。其优点是方便查找，缺点是插入慢。



**LinkedList**：内部实现方式是基于双向链表实现的，并且有头尾节点，其优点是插入方便，缺点是查找慢。



**Vector**：内部实现方式同ArrayList，继承的接口和实现的类与ArrayList相同，而且都是基于数组实现的，与ArrayList的区别是，Vector是线程安全的。默认初始化长度为10，扩容时，如果capacityIncrement为0的话，新的长度为原来长度的二倍，如果大于0,新长度为原来的长度加上capacityIncrement。常用的方法同ArrayList。

ArrayList适合遍历和查找，LinkedList适合插入和删除。

1. set

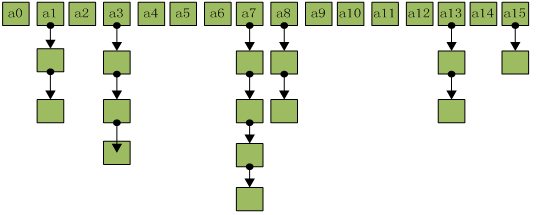
**HashSet**：其特点是不可重复，无序，可以存放一个null值，其内部实现是基于HashMap实现的，利用key的不可重复，value对应的类型为object,不保存值。

**TreeSet**：其实现方式是基于TreeMap进行实现的，利用TreeMap key的不可重复。

**LinkedHashSet**：其具有set的特点，但也能够保持迭代顺序为访问顺序或插入顺序，内部实现是基于LinkedHashMap实现的。

1. Map

**HashMap**:存放key-value类型数据，HashMap内部实现是基于数组加单向链表，数组长度初始化长度为16，长度最大值为2的30次方，首先会生成key的hash值，然后对hash值按照数组的长度进行取余，如果该数组对应的链表已经有值了，将会插入到该链表的末尾，jdk1.8中对链表进行了优化，当链表的长度大于8时，会将链表转换成红黑树，当对数组进行扩容时，如果数组元素对应的红黑树数量少于6时，会将其转化成链表，否则的话仍然是红黑树，当数组的长度大于64时，才会进行树化，小于64时，将首先考虑的是扩大数组的长度。Key可为null，但仅可以有一个，value可以为null值。使用HashMap时，可以预估下数据的可能长度，尽量避免扩容，因为扩容会造成效率很慢。



**LinkedHashMap**:LinkedHashMap是继承于HashMap的，在HashMap之上维护了一个双向链表，这个双向链表是继承于HashMap的Entry，在HashMap之上增加了before和after，HashMap保存时，不会考虑插入时候的顺序，LinkedHashMap遍历时候，会将能够按照插入的顺序进行读取。其顺序保持有两种方式，一种是根据插入顺序来访问，一种是依据访问顺序，对应的方法是put(),和get()方法。LinkedHashMap在put()时候，首先调用HashMap的put()方法，然后再维护在双向链表中的关系。

<https://www.jianshu.com/p/8f4f58b4b8ab>

**TreeMap**：同HashMap，TreeMap也是保存key-value的，但基于红黑树来实现的,红黑树的特点是节点要么是红色要么是黑色，根节点必须是黑色，空叶子节点为黑色，红色节点的子节点必须为黑色，从一个节点到孙节点的黑色节点数目必须相同，为什么有红黑树了，这是为了保证二叉树极端情况下成为线性链。红黑树具有查找、删除、插入的时间复杂度为logn的特点。其节点为Entry类型，包含的key value left right parent color。

HashTable:其实现类似HashMap，与HashTable的主要区别是其是线程安全的。其扩容为oldCapatity << 1 + 1 ,hashtable不允许key和value为null

**ConcurrentHashMap**:juc中的一个并发处理的Map容器，其主要区别是和hashtable的区别，hashtable是线程安全的，但其性能非常差，是通过对整个map进行加锁，当多个线程进行读取map内的内容时，会被阻塞，相当于操作串行化，concurrentHashMap的思想是进行分段加锁，也就是segment,segments继承可重入锁。他有三个参数，initialCapacity(初始化容量)，loadfactor（负载因子），concurrentLevel(默认为16)，为什么是2的次幂，来定位index时候，比较方便。每个segment由segmentshift和segmentmask来定位。

<https://www.cnblogs.com/chengxiao/p/6842045.html>

hashmap和concurrenthashmap的区别：

<http://www.codeceo.com/article/java-hashmap-concurrenthashmap.html>