

# Container

## 1. java 容器都有哪些？

- Collection: 服从特定规则的元素序列。包含List, Set, Queue, Iterable;
- Map: "键值对"对象，运行按照key查找value。包含HashTable, HashMap, SortedMap。

## 2. Collection 和 Collections 有什么区别？

- Collection是一个集合接口，提供了对集合对象进行基本操作的通用接口方法。
- Collections是一个集合类，包含了各种集合操作的静态多态方法，无法被实例化，类似工具类，服务于Collection框架。

## 3. List、Set、Map 之间的区别是什么？

- List: 元素有序、有重复；
- Set: 元素无序、可重复；
- Map: k-v对存储，k无序不可重复的，v无序可重复的，每个k对应唯一v。

## 4. HashMap 和 Hashtable 有什么区别？

- HashMap非线程安全，HashTable方法都加了synchronized锁所以是线程安全的；
- HashMap允许存储null的k和v，后者不可以；
- 初始容量和扩容机制不同：Map初始16，扩容2n，Table初始11，扩容2n+1；给定初始容量Map给的2的幂次方大小。
- 底层数据结构：Map在jdk1.8之后在链表长度大于8时，转换成红黑树进行查找。
- HashTable效率低下，基本被弃用了。

## hashmap为什么不是线程安全的

- 扩容时需要resize，其中调用transfer会因为链表采用头插法而使得转移后链表逆序，从而产生死锁；
- 多线程rehash时会造成环形链表
- fail-fast: 遍历map的过程中修改map中值导致ConcurrentModificationException

## 5. 如何决定使用 HashMap 还是 TreeMap？

TreeMap相较于HashMap，因为它继承了NavigableMap接口，所以可以对集合内元素进行搜索，同时还继承了SortedMap接口， 所以可以对集合内元素按照k排序。类似集合类排序方法Collections.Sort()中重写Comparator实现自定义排序类似，当我们需要 对Map中元素进行自定排序和查询的时候就有限选择TreeMap。

## 6. 说一下 HashMap 的实现原理？

底层数据结构是 **数组+链表** 结合在一起，也就是链表数组。HashMap通过对key的hashcode进行扰动处理之后得到hash值，然后通过 $(n-1) \& \text{hash}$ 来确定元素存放的位置（n为数组长度）， 如果当前位置存在元素，则需要判断冲突元素的hash值和key是否相同，相同则覆盖，否则拉链法解决冲突。jdk1.8之后解决hash冲突：链表长度 $>8$ ，先判断数组长度 $<64$ ，是则先扩容，否则将链表转化为红黑树优化搜索效率。

- **扰动函数**：就是HashMap的hash方法，目的是为了减少碰撞（hashcode相同的情况）
- **拉链法**：遇到冲突，将冲突值加到对应数组位置的链表中即可
- **红黑树**：为了解决二叉查找树在极端情况下退化成线性链表结构，严格保证查找在logN复杂度

## 7. 说一下 HashSet 的实现原理？

HashSet 底层就是基于 HashMap 实现的。关键点就是在于HashSet是如何去重的： 和HashMap类似，先计算hash值确定存放位置；然后如果发现相同hashcode的对象，会调用equals方法判断对象是否真的相，如果相同，则不会加入成功。

## 8. ArrayList 和 LinkedList 的区别是什么？

- 都不是线程安全的；
- 底层数据结构：前者是Object数组（动态数组），后者是双向链表；
- 插入和删除：前者由于是数组受位置影响，后者链表不受影响为O(1)；
- 查找：数组直接去下标可以快速随机访问，链表不可以；
- 内存占用：前者由于是list所以尾部要预留一定空间，链表每个元素都要留空间存前驱和后继结点信息等。

## 9. ArrayList 和 Vector 的区别是什么？

ArrayList 是 List 的主要实现类，底层使用 Object[] 存储，适用于频繁的查找工作，线程不安全； Vector 是 List 的古老实现类，底层使用 Object[] 存储，线程安全的。

## 10. 在 Queue 中 poll()和 remove()有什么区别？

都是返回并删除队列以第一个元素，队列为空poll会返回null， remove会抛出NoSuchElementException。

## 11. 哪些集合类是线程安全的？

HashTable, Stack, Vector以及java.util.concurrent包下所有的集合类

## 12. 迭代器 Iterator 是什么？怎么使用？有什么特点？

迭代器是用来遍历集合的，提供一种方法访问容器中元素而不需要暴露该对象的内部细节。特点是更安全，相比较于foreach等遍历方式，在遍历过程中集合元素被更改会抛出ConcurrentModificationException 使用就是hasNext()的话用next()获取下一个元素

## 13. 怎么确保一个集合不能被修改？

- 利用Collections和Guava提供的类可实现的不可变对象：
- Collections.unmodifiableXXX:Collection、List、Set、Map...
- GuavaImmutableXXXCollection、List、Set、Map...