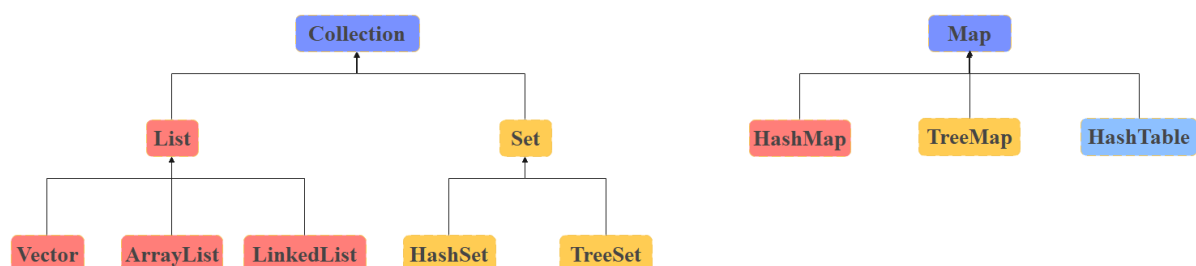


【第01话：面试官问起容器，千万不要以为是Spring 容器】

月薪1万的小明同学是如何回答的



具体文字回答：

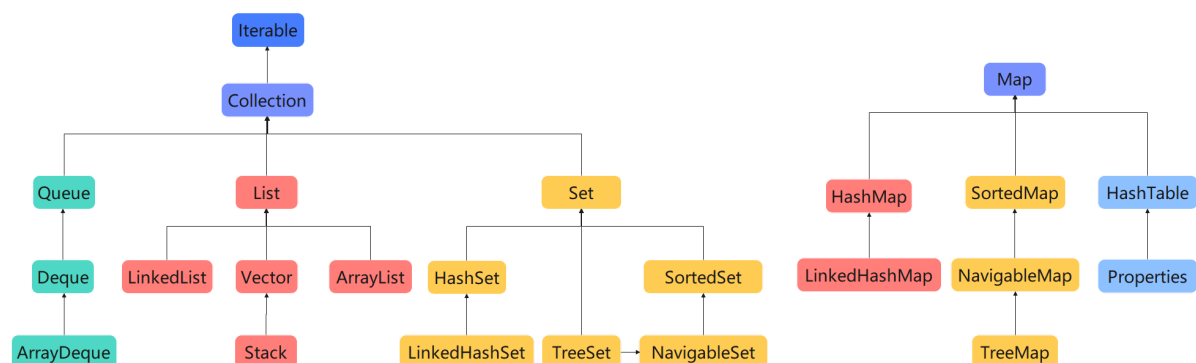
Java中的容器就是集合。在集合分为两个顶级接口，分别是Collection接口和Map接口。Collection接口有两个子接口，分别是List用来存储有序可重复数据和Set用来存储无序不可重复数据。

List接口的具体实现包含Vector、ArrayList、LinkedList。其中Vector基于synchronized关键字实现的线程安全可变长度数组。ArrayList底层是基于数组扩容实现的非线程安全可变长度数组容器。LinkedList底层是双向链表。

Set接口具体实现包含HashSet和TreeSet。HashSet底层是基于HashMap实现的。TreeSet底层是基于TreeMap实现的。整个Set接口的实现基本都是基于Map接口实现的。

Map接口常用实现类包含HashMap、TreeMap、HashTable。其中HashMap底层是数组+链表+红黑树。TreeMap底层是红黑树。HashTable和HashMap类似，但是基于synchronized保证了线程安全。

月薪1万5的小明同学是如何回答的



具体文字回答：

Java中的容器就是集合。在集合中分为两大接口：分别是Collection接口和Map接口。

Collection接口不是顶级接口，它是Iterable接口的子接口。Iterable接口表示可迭代的。也就是说Collection接口所有子接口和实现类都是可以通过Iterator迭代器进行迭代。在Collection接口下有三个子接口，分别是Queue、List、Set。

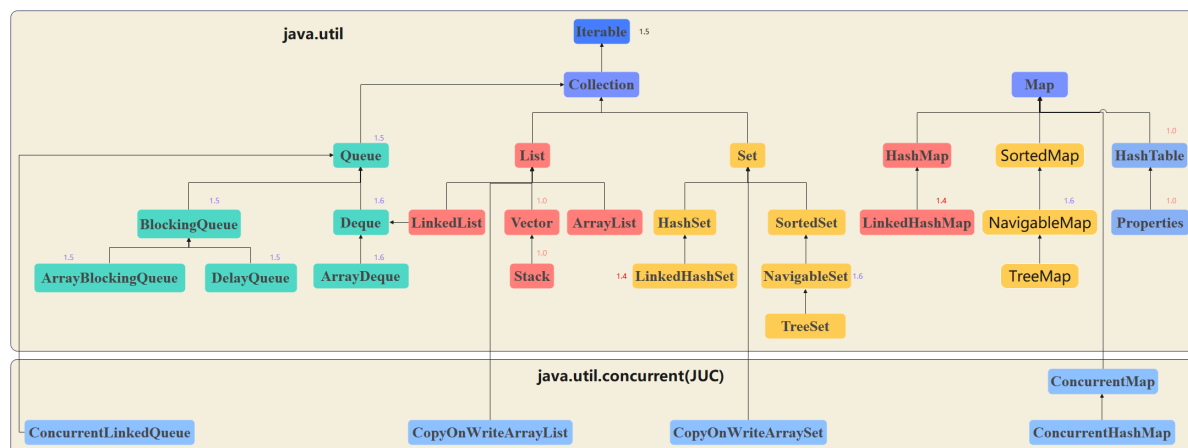
Queue接口表示队列，Queue的子接口Deque表示双向队列。Deque的具体实现类是ArrayDeque，表示基于数组的队列实现。当需要使用队列数据结构时可以考虑使用ArrayDeque。

List接口用来存储有序可重复数据。具体实现类有LinkedList、ArrayList、Vector。LinkedList底层是基于双向非循环链表实现的，适合用在增加、删除远远大于查询的场景。ArrayList底层是基于数组实现的，遍历效率较高。Vector和ArrayList类似，使用synchronized保证线程安全，但是在不考虑线程安全的情况下，很少使用Vector。Vector还有一个子类Stack，用来表示栈。所以当考虑使用栈数据结构时，可考虑Stack。

Set接口所有的实现类和子接口都和Map接口对应。包含HashSet、TreeSet、子接口SortedSet。HashSet是基于HashMap实现的，把HashSet的值作为HashMap的key，存储一个Object对象作为HashMap的Value。HashSet还有一个子类LinkedHashSet，一个基于LinkedHashMap的类。子接口SortedSet表示可排序集合，SortedSet还有一个子接口NavigableSet表示可做范围查询的集合。TreeSet即实现了Set接口，又实现了NavigableSet，底层是TreeMap。

Map接口有两个实现类HashMap、HashTable和一个子接口SortedMap。HashMap底层是基于数组+链表+红黑树实现的，在需要快速获取容器中一个值的场景中使用频率比较高，是平时开发时非常常用的集合。HashMap还有个子类LinkedHashMap，在一些JSON数据转换时默认转换类型就是它。HashTable和HashMap类似，但是有synchronized，每次操作会锁住整个HashTable，效率较低，目前使用较少。HashTable的子类Properties常用在属性文件读取上。SortedMap表示可对Key做排序，子接口NavigableMap表示可做范围查询，具体实现类是TreeMap，底层是红黑树。

月薪2万+的小明同学是如何回答的



Java中容器就是集合。集合中分为两个部分，分别是java.util包下的集合和java.util.concurrent包下的集合。

整个Java中集合中常用的接口和类主要集中在1.0版本、1.2版本、1.4版本、1.5版本和1.6版本。1.7和1.8甚至之后版本中出现的接口和类使用频率并不多。

在Java1.0版本中出现的是Vector和子类Stack，HashTable和子类Properties。其中Vector是线程安全的可变长度数组。除方法带有synchronized，其他基本与ArrayList类似。在不考虑线程安全情况下，由于其性能问题，使用较少。从Java 5开始出现了CopyOnWriteArrayList，里面很多线程安全都是基于CAS乐观锁的，性能更好。所以Vector在当下使用频率也不是特别高。

Stack代表栈。类中提供了基本的入栈操作和出栈操作。在希望使用栈时，偶尔能使用到。

HashTable和后面的1.2版本中出现的HashMap类似。与HashMap的主要区别是：Key不允许出现null值，是synchronized的。由于每次修改数据都需要锁住整个HashTable，导致效率比较低。所有在不需要考虑线程安全情况下，更多选择HashMap。而从Java 5开始出现ConcurrentHashMap，属于HashTable的升级版，所以HashTable在目前使用的频率也不太高了。

Properties在目前项目中还是能用到的。用来存储属性值。

从1.2版本开始，Java容器是按照面向接口编程。整个集合体系中使用大量接口定义方法。包含Collection接口、Map接口及子接口和实现类。Collection接口最开始是顶级接口，后来到了1.5版本后出现了Iterable接口，表示数据可使用迭代器迭代，才把Collection作为Iterable的子接口。

Collection出现时同时还有两个子接口List和Set。并把LinkedList和ArrayList作为List接口的实现。到目前为止基于链表的LinkedList和ArrayList使用频率依然非常高。例如持久层数据都是用ArrayList存储。为了规范整个集合，把Vector也作为了List接口的实现类。

Set接口刚出现时有目前非常常用的HashSet和实现类SortedSet的TreeSet。Set接口就是标准Map接口的变形，Set的值都是Map中的Key。

Map刚出现时包含HashMap和子接口SortedMap接口的实现类TreeMap。HashMap就是Java帮助对散列表的实现，为了防止Hash碰撞，使用了链式法。到了1.8版本后又添加了红黑树在少量数据时提升查询效率。同样是为了规范类把HashTable作为了Map的实现类。

到了1.4版本出现HashMap子类，LinkedHashMap，底层使用双向链表存储所有Entry。

在后来1.5版本出现了List接口子接口队列Queue，和高并发场景中的类。

Queue里面包含子接口BlockQueue表示阻塞队列。同时还有基于数组的ArrayBlockingQueue和延迟队列DelayQueue。

JUC中ConcurrentLinkedQueue：是JUC中提供的基于链表的无界队列实现。CopyOnWriteArrayList：属于ArrayList的并发支持类。底层是基于读写分离思想实现的，适用于读远远大于写的场景。

CopyOnWriteArraySet：属于HashSet的并发支持类，但其底层是基于CopyOnWriteArrayList实现的。

ConcurrentHashMap：在各大框架底层都能看见的一个类，属于HashMap在并发场景中可以使用的类。

到了1.6版本主要是对Queue添加了子接口Deque，表示双向队列，及Deque的实现类，基于数组的双向队列ArrayDeque。同时还让LinkedList实现了Deque，表示基于链表的双向队列。

还有就是对SortedMap添加了子接口NavigableMap做范围查询的，TreeMap不再直接实现SortedMap，而是实现NavigableMap。TreeSet也是相同道理，多加了一个父接口NavigableSet。

以上就是Java中容器。