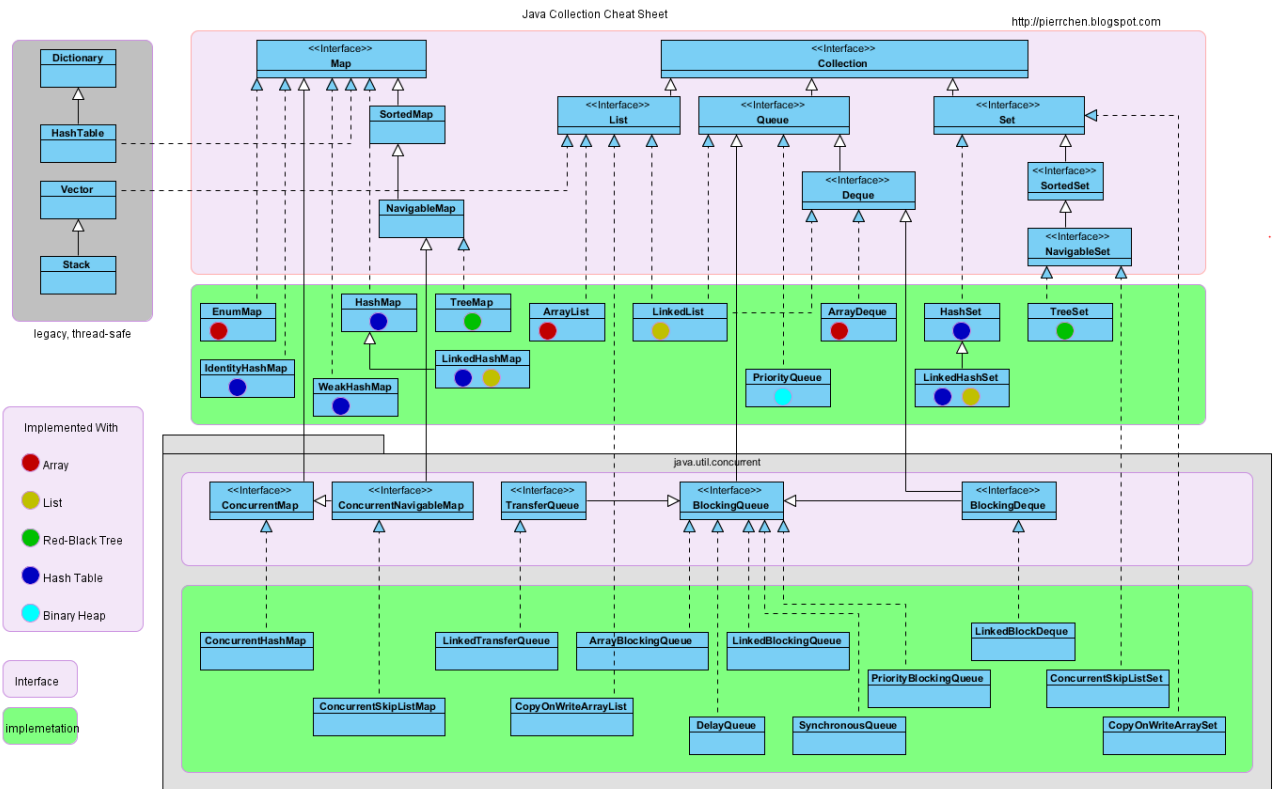


Collection 类关系图

知识体系结构



介绍

容器，就是可以容纳其他Java对象的对象。*Java Collections Framework(JCF)*为Java开发者提供了通用的容器，其始于JDK 1.2，优点是：

- 降低编程难度
- 提高程序性能
- 提高API间的互操作性
- 降低学习难度
- 降低设计和实现相关API的难度
- 增加程序的重用性

Java容器里只能放对象，对于基本类型(int, long, float, double等)，需要将其包装成对象类型后(Integer, Long, Float, Double等)才能放到容器里。很多时候拆包装和解包装能够自动完成。这虽然会导致额外的性能和空间开销，但简化了设计和编程。

Collection

容器主要包括 Collection 和 Map 两种，Collection 存储着对象的集合，而 Map 存储着键值对(两个对象)的映射表。

Set

TreeSet

基于红黑树实现，支持有序性操作，例如根据一个范围查找元素的操作。但是查找效率不如 HashSet，HashSet 查找的时间复杂度为 $O(1)$ ，TreeSet 则为 $O(\log N)$ 。

HashSet

基于哈希表实现，支持快速查找，但不支持有序性操作。并且失去了元素的插入顺序信息，也就是说使用 Iterator 遍历 HashSet 得到的结果是不确定的。

LinkedHashSet

具有 HashSet 的查找效率，且内部使用双向链表维护元素的插入顺序。

List

ArrayList

基于动态数组实现，支持随机访问。

Vector

和 ArrayList 类似，但它是线程安全的。

LinkedList

基于双向链表实现，只能顺序访问，但是可以快速地在链表中间插入和删除元素。不仅如此，LinkedList 还可以用作栈、队列和双向队列。

Queue

LinkedList

可以用它来实现双向队列。

PriorityQueue

基于堆结构实现，可以用它来实现优先队列。

Map

TreeMap

基于红黑树实现。

HashMap

基于哈希表实现。

HashTable

和 HashMap 类似，但它是线程安全的，这意味着同一时刻多个线程可以同时写入 HashTable 并且不会导致数据不一致。它是遗留类，不应该去使用它。现在可以使用 ConcurrentHashMap 来支持线程安全，并且 ConcurrentHashMap 的效率会更高，因为 ConcurrentHashMap 引入了分段锁。

LinkedHashMap

使用双向链表来维护元素的顺序，顺序为插入顺序或者最近最少使用(LRU)顺序。