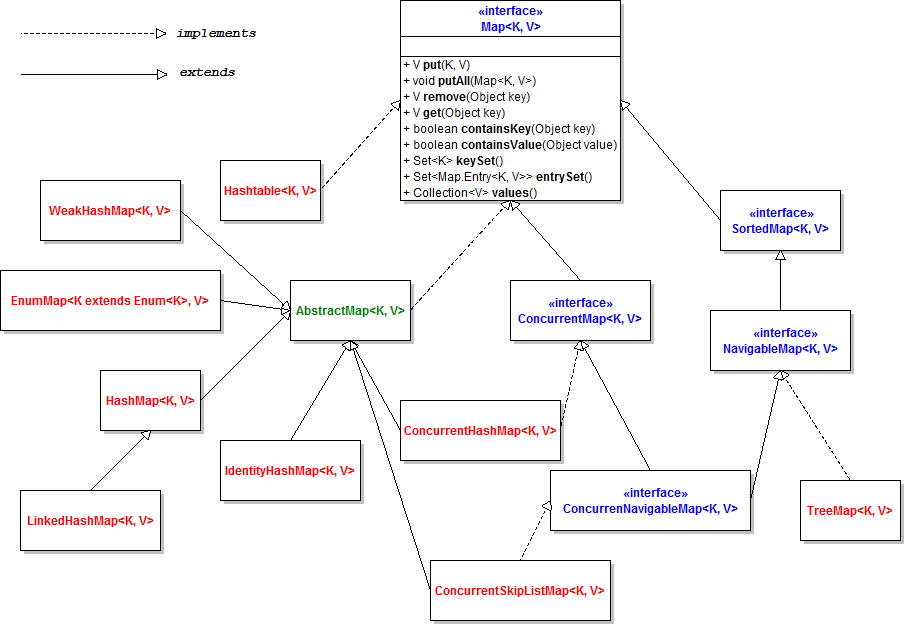


В библиотеке коллекций Java существует два базовых интерфейса, реализации которых и представляют совокупность всех классов коллекций:

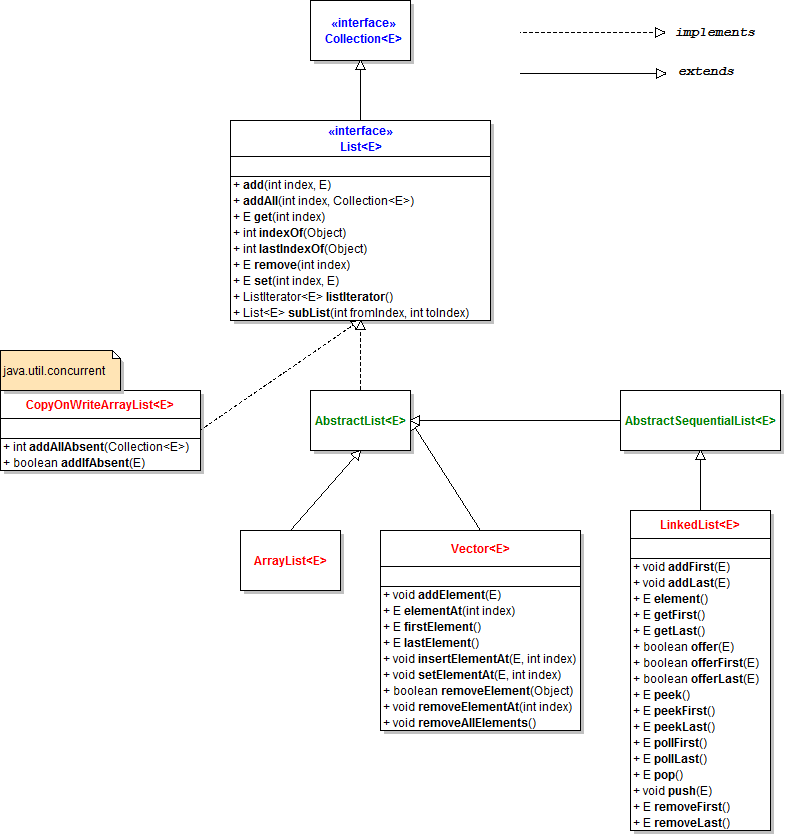
1. **Collection**- коллекция содержит набор объектов (элементов). Здесь определены основные методы для манипуляции с данными, такие как вставка (add, addAll), удаление (remove, removeAll, clear), поиск (contains)  
2. **Map**-  описывает коллекцию, состоящую из пар "ключ — значение". У каждого ключа только одно значение, что соответствует математическому понятию однозначной функции или отображения (тар). Такую коллекцию часто называют еще словарем (dictionary) или ассоциативным массивом (associative array). Никак НЕ относится к интерфейсу Collection и является самостоятельным.

**Map**



1. Hashtable хештаблица, не может содержать Null и является synchronized(все методы)
2. hashMap хештаблица.
3. LinkedHashMap хештаблица, итератор будет в порядке добавления(все элементы в связный список).
4. TreeMap Хранит элементы в красно-черном дереве(по ключу). Итератор тоже отсортирован.
5. WeakHashMap если ссылок больше нет на ключ, то элемент будет удален из карты.

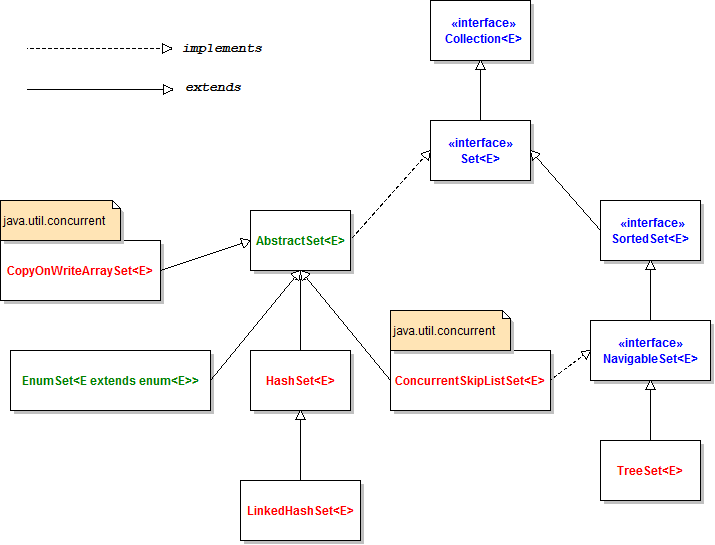
**List**



**List**- Представляет собой неупорядоченную коллекцию, в которой допустимы дублирующие значения. Иногда их называют последовательностями (sequence ). Элементы такой коллекции пронумерованы, начиная от нуля, к ним можно обратиться по индексу.

1. ArrayList внутри массив. Когда место заканчивается создается новый и всё копируется. При удалении всё сдвагается.
2. **LinkedList**- Двусвязный список. Скорость доступа-линейная.
3. Vector. Как ArrayList, только синхронный.

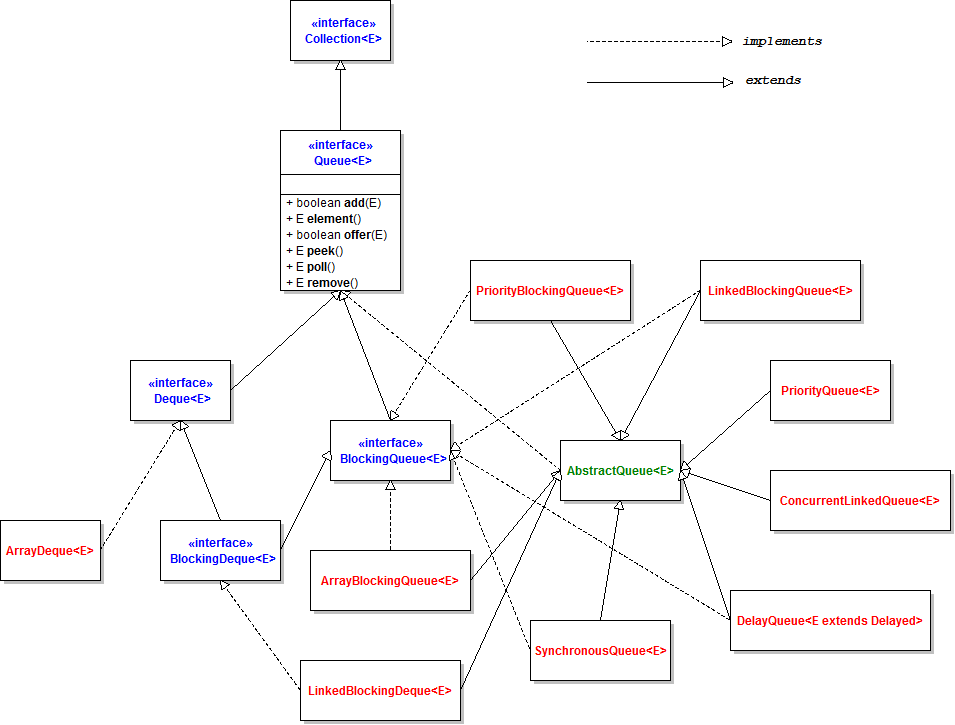
**Set**

****

. **Set**- описывает неупорядоченную коллекцию, не содержащую повторяющихся элементов. Это соответствует математическому понятию множества (set).

1. **HashSet. Внутри HashMap. Ключ-добавляемое значение, значение – пустой объект.**
2. LinkedHashSet. **Внутри LinkedHashMap**
3. TreeSet. содержит в себе объект NavigableMap
4. EnumSet. Представляет внутри битовую последовательность где 0 нет элемента из Enum, а 1 – значит есть. Привязана жестко к определенному ENUm.

**Queue**

****

**Queue**- очередь. Сразу запоминаем как правильно произносится: Queue - КЬЮ Это коллекция, предназначенная для хранения элементов в порядке, нужном для их обработки. В дополнение к базовым операциям интерфейса Collection, очередь предоставляет дополнительные операции вставки, получения и контроля.

1. **PriorityQueue хранит элементы в заданному Comparator порядке.**
2. [**ArrayDeque**](http://docs.oracle.com/javase/8/docs/api/java/util/ArrayDeque.html)двухсторонняя очередь.



1. f(x)=O(g(n)). Здесь темпы роста не быстрее, чем g (n). O большое является наиболее полезной, поскольку представляет наихудший случай.

http://habrahabr.ru/post/188010/