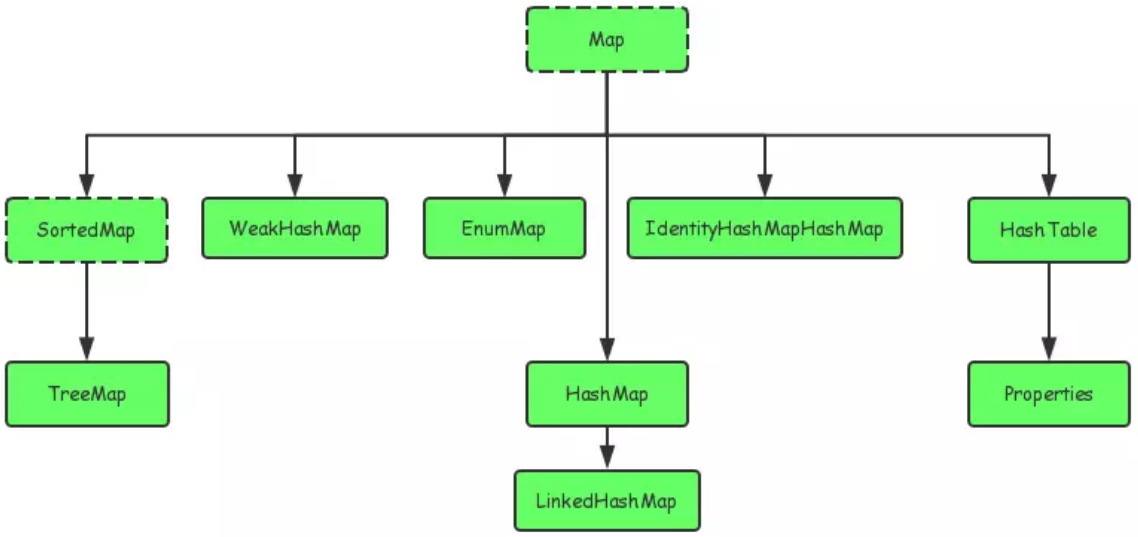
Map是一个映射接口，用来存储“键(key)-值(value) 对”，存储无序。 key和value都可以是任何引用类型的数据。



- TreeMap：基于红黑树实现。
- HashMap：基于哈希表实现。
- HashTable：和 HashMap 类似，但线程安全，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。

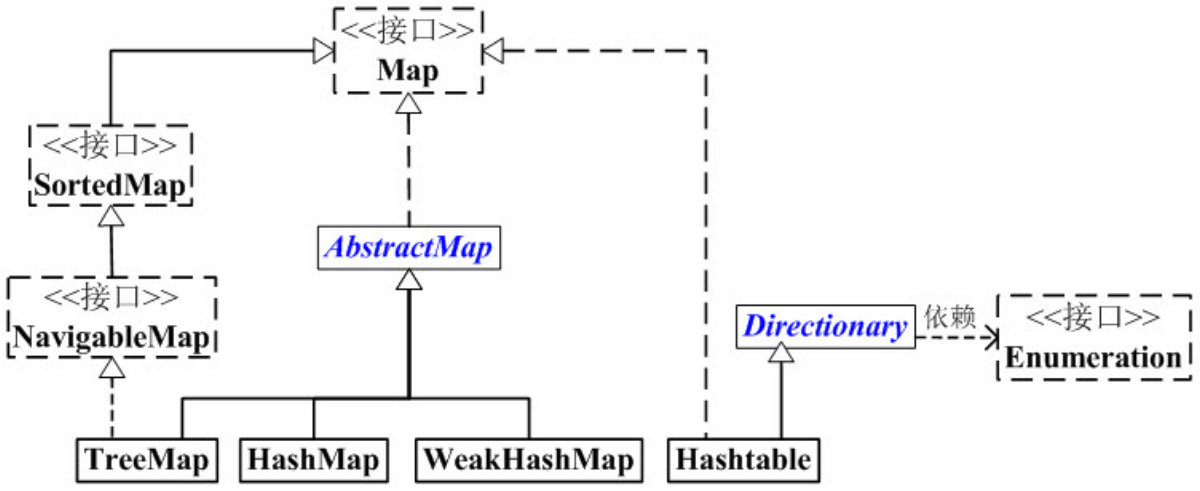
AbstractMap是个抽象类，它实现了Map接口中的大部分API。

SortedMap 是继承于Map的接口。SortedMap中的内容是**排序的键值对**，排序的方法是通过比较器(Comparator)。

NavigableMap 是继承于SortedMap的接口。相比于SortedMap，NavigableMap有一系列的导航方法；如"获取大于/等于某对象的键值对"、“获取小于/等于某对象的键值对”等等。

Hashtable 不继承于AbstractMap，但它继承于Dictionary(Dictionary也是键值对的接口)，也实现Map接口。因此，Hashtable的内容也是“**键值对，也不保证次序**”。但和HashMap相比，Hashtable是线程安全的，而且它支持通过Enumeration去遍历。

WeakHashMap 继承于AbstractMap。它和HashMap的键类型不同，WeakHashMap的键是**“弱键”**。



Map接口

Map 接口提供三种collection 视图，允许以**键集**、**值集**或**键-值映射关系集**的形式查看某个映射的内容。映射顺序有些实现类可以确保。

Map 的实现类应该提供2个“标准的”构造方法：**第一个，void（无参数）构造方法，用于创建空映射**；**第二个，带有单个 Map 类型参数的构造方法，用于创建一个与其参数具有相同键-值映射关系的新映射。**实际上，后一个构造方法允许用户复制任意映射，生成所需类的一

个等价映射。尽管无法强制执行此建议（因为接口不能包含构造方法），但是 JDK 中所有通用的映射实现都遵从它。

表9-3 Map接口中常用的方法

方法	说明
Object put(Object key, Object value)	存放键值对
Object get(Object key)	通过键对象查找得到值对象
Object remove(Object key)	删除键对象对应的键值对
boolean containsKey(Object key)	Map容器中是否包含键对象对应的键值对
boolean containsValue(Object value)	Map容器中是否包含值对象对应的键值对
int size()	包含键值对的数量
boolean isEmpty()	Map是否为空
void putAll(Map t)	将t的所有键值对存放到本map对象
void clear()	清空本map对象所有键值对

entrySet()用于返回**键-值集**的**Set集合**

keySet()用于返回**键集**的**Set集合**

values()用户返回**值集**的**Collection集合**

因为Map中不能包含重复的键；每个键最多只能映射到一个值。所以，**键-值集、键集都是Set，值集时Collection**。Map.Entry是Map中内部的一个接口，封装了一个key-value对（在HashMap实现为单向链表）。Entry包含如下方法：

boolean	**equals**(Object o) 比较指定对象与此项的相等性。
K	**getKey**() 返回与此项对应的键。
V	**getValue**() 返回与此项对应的值。
int	**hashCode**() 返回此映射项的哈希码值。
V	**setValue**(V value) 用指定的值替换与此项对应的值（可选操作）。

SortedMap

是一个有序的键值映射。排序方式：**自然排序**或者**用户指定比较器。**插入有序 SortedMap 的所有元素都必须实现 Comparable 接口（或者被指定的比较器所接受）。

所有SortedMap 实现类都应该提供 4 个“标准”构造方法：

(01) **void（无参数）构造方法**，它创建一个空的有序映射，按键的自然顺序进行排序。

(02) **带有一个 Comparator 类型参数的构造方法**，它创建一个空的有序映射，根据指定的比较器进行排序。

(03) **带有一个 Map 类型参数的构造方法**，它创建一个新的有序映射，其键-值映射关系与参数相同，按照键的自然顺序进行排序。

(04) **带有一个 SortedMap 类型参数的构造方法**，它创建一个新的有序映射，其键-值映射关系和排序方法与输入的有序映射相同。无法保证强制实施此建议，因为接口不能包含构造方法。

NavigableMap

是一个可导航的键-值对集合，具有了为给定搜索目标报告最接近匹配项的导航方法。额外提供的功能可以分为4类：

第1类，**提供操作键-值对的方法。**

lowerEntry、floorEntry、ceilingEntry 和 higherEntry 方法，它们分别返回与小于、小于等于、大于等于、大于给定键的键关联的Map.Entry对象。firstEntry、pollFirstEntry、lastEntry 和 pollLastEntry 方法，它们返回和/或移除最小和最大的映射关系（如果存在），否则返回 null。

第2类，**提供操作键的方法。**和第1类类似

lowerKey、floorKey、ceilingKey 和 higherKey 方法，它们分别返回与小于、小于等于、大于等于、大于给定键的键。

第3类，**获取键集。**

navigableKeySet、descendingKeySet分别获取正序/反序的键集。

第4类，获取键-值对的子集。

Dictionary

JDK 1.0定义的键值对的接口，它也包括了操作键值对的基本函数。