# TreeSet

Treset:内部使用TreeMap来存放数据，不过数据是存放在key中的，value是自定义的一个Object对象

// 存放数据的Map对象--》TreeMap

**private** **transient** NavigableMap<E, Object> m;

// Treeset将数据存放在Map的key中，PRESENT作为value，就是用来占据value位置的没有实际意义

**private** **static** **final** Object PRESENT = **new** Object();

构造函数

1. 无参构造函数

**public** MyTreeSet()

{

**this**(**new** MyTreeMap<E, Object>());创建一个TreeMap对象

}

1. 定义一个比较器（实现啊Comparator接口）作为参数的构造函数

**public** MyTreeSet(Comparator<? **super** E> comparator)

{

**this**(**new** MyTreeMap<>(comparator));//创建一个传入比较器的treeMap对象

}

1. 将一个Map对象作为参数的构造函数

MyTreeSet(NavigableMap<E, Object> m)

{

**this**.m = m;map赋值

}

1. 集合作为参数的构造函数

**public** MyTreeSet(Collection<? **extends** E> c)

{

**this**();

addAll(c);

}

# HashSet

HashSet是通过hashMap来保存数据的，内存定义hashMap,将存储的值作为hashMap的key,保证了数据不会有重复的

# LinkedHashSet

内部存储数据的是LinkedHashMap，数据存放在key中，保证数据不会有重复的