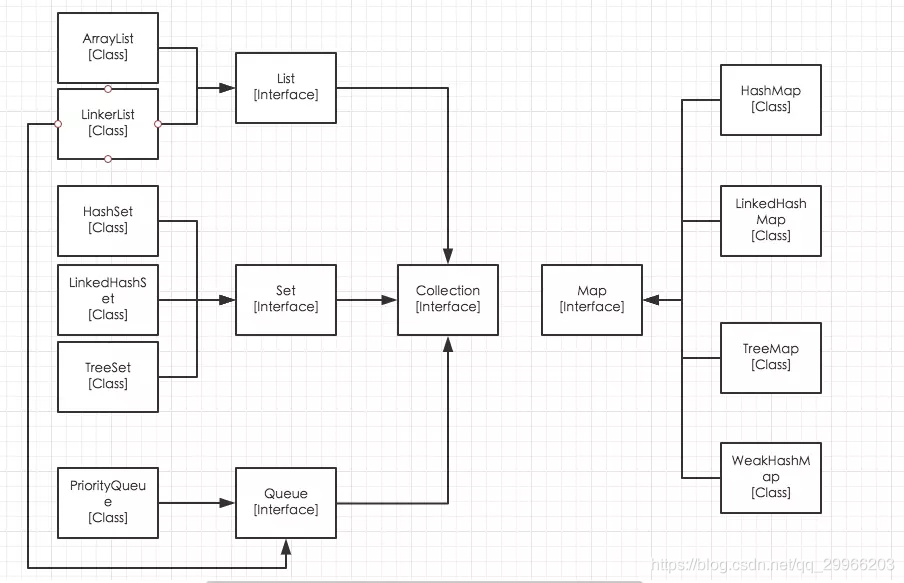
# Java 集合框架

## **Java 集合类继承关系**



# Java 集合类简介

### **Collection （Interface）**

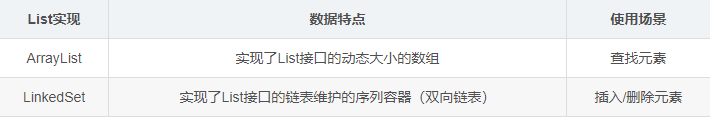
Collection是最基本的集合接口，存储对象元素集合。一个Collection代表一组Object（元素）。  
适用Iterator迭代器遍历集合元素



List （Interface）

List是一个允许重复元素的指定索引、有序集合。集合中每个元素都有其对应的顺序索引。List集合允许使用重复元素，可以通过索引来访问指定位置的集合元素 。List集合默认按元素的添加顺序设置元素的索引，例如第一个添加的元素索引为0，第二个添加的元素索引为1…

从List接口的方法来看，List接口增加了面向位置的操作，允许在指定位置上插入/访问元素。



Set （Interface）

Set是一种不包含重复的元素的Collection，即任意的两个元素e1和e2都有e1.equals(e2)=false，Set最多有一个null元素。

如果试图把两个相同的元素加入同一个Set集合中，则添加操作失败，add()方法返回false，且新元素不会被加入。



Set/HashSet如何确保它的唯一性？

* 向Set中添加元素时需先判断是否有相同key的元素

1. 首先判断set集合中是否有与新添加数据hashcode值一致的元素
2. 如果有hashcode相同的元素，再调用equals方法进一步判断

因此，在Java运行时环境判断HashSet和HastMap中的两个对象相同或不同应该先判断hashcode是否相等，再判断是否equals。 只有两者均相同，才能保证对象的一致性。

为了保证HashSet中的对象不会出现重复值，在被存放元素的类中必须要重写hashCode()和equals()这两个方法。

重写规范

1. 如果两个对象相同，那么他们的hashcode应该相等
2. 如果两个对象不相同，他们的hashcode可能相同

Queue （Interface）

Queue用户模拟队列这种数据结构，队列通常是指“先进先出”(FIFO，first-in-first-out)的容器。队列的头部是在队列中存放时间最长的元素，队列的尾部是保存在队列中存放时间最短的元素。新元素插入（offer）到队列的尾部，访问元素（poll）操作会返回队列头部的元素。通常，队列不允许随机访问队列中的元素。

### **Map （Interface）**

Map是图接口，存储键值对映射的容器类。Map提供key到value的映射。一个Map中不能包含相同的key，每个key只能映射一个value。



* Map的遍历方式  
  Map集合提供3种遍历访问方法，

1. Set keySet() 获得所有key的集合然后通过key访问value
2. Collection values() 获得value的集合
3. Set< Map.Entry< K, V>> entrySet() 获得key-value键值对的集合

Entry<K,V>为Map<K,V>的一个内部接口，其实每一个键值对都是一个Entry的实例关系对象，所以Map实际其实就是Entry的一个Collection。