



链滴

java 集合总结

作者: [CoderZhang](#)

原文链接: <https://ld246.com/article/1681178263361>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

先上一张集合家族的结构图，对集合框架有一个整体的认识：

本文出处：<https://my.oschina.net/liuyuantao/blog/1512252>

<https://my.oschina.net/ouyangsihai/blog/1806516>

对其中的重点进行总结，这样有利于理解与记忆

对于在集合中常用到的泛型：

泛型：就是类型的参数化，泛型是类型的一部分，泛型+类名是一个整体。

泛型的通配符(?)：

上限限定：比如定义方法的时候出现，`public void getFunc(List<? extends Animal> an)`，那么表示这里的参数可以传入Animal，或者 Animal的子类

下限限定：比如定义方法的时候出现，`public void getFunc(Set<? super Animal> an)`，那么表示这里的参数可以传入Animal，或者Animal的父类

使用泛型的注意点：

- 1、泛型不支持基本数据类型
- 2、泛型不支持继承，必须保持前后一致（比如这样是错误的：`List<Object> list = new ArrayList<String>()`;
- 3、泛型如果不指定具体类型，参数会自动提升我Object 类型。

Collection 体系：

1.Collection 的主要体系：List、Set

- List：存取有序有索引，元素可重复，主要实现类有：ArrayList, LinkedList, Vector(已过时)
- set：存取无序，元素不可以重复，主要实现类有：HashSet, LinkedHashSet, TreeSet

2.List 体系总结

● ArrayList:底层是数组结构，查询快，增删慢。线程不安全，效率高。由于ArrayList 底层是维护一个可变数组，说以在所以若知道我们需要存储数据的容量前提下，创建对象时指定list 的大小，可免底层数组频繁扩容，提高性能。

● LinkedList:是基于链表结构实现的，查询速度慢，增删速度快，提供了对头尾元素进行增删查方法，通过对头尾元素增删查的的先后顺序实现实现栈和队列；栈是先进后出，而队列是先进先出

3.Set 体系：

● HashSet: 哈希表是通过hashCode和equals方法来共同保证元素的唯一性，所以对自定义对象保证唯一性，一定要重写这两个方法。

哈希表的存储数据过程（哈希表底层也维护了一个数组）：

根据存储的元素计算出hashCode值，然后根据计算得出的hashCode值和数组的长度计算出存储的标；如果下标的位置无元素，那么直接存储；如果有元素，那么使用要存入的元素和该元素进行equal方法，如果结果为真，则已经有相同的元素了，所以直接不存；如果结果假，那么进行存储，以链表形式存储。所以注意点：在重写hashCode方法和equals方法时要保证比较结果的一致性。

- `LinkedHashSet`:是基于链表和哈希表共同实现的，所以具有存取有序，元素唯一的特点
- `TreeSet`:存取无序，元素唯一，存储过程中会进行排序，`TreeSet`是基于二叉树的数据结构

二叉树的存储过程：

如果是第一个元素，那么直接存入，作为根节点，下一个元素进来是会跟节点比较，如果大于节点放边的，小于节点放左边；等于节点就不存储。后面的元素进来会依次比较，直到有位置存储为止，

`TreeSet`保证元素的唯一性是有两种方式**： **

- 1、自定义对象实现`Comparable`接口，重写`compareTo`方法，该方法返回0表示相等，小于0表示准存入的元素比被比较的元素小，否则大于0；
- 2、在创建`TreeSet`的时候向构造器中传入比较器`Comparator`接口实现类对象，实现`Comparator`接重写`compara`方法。

如果向`TreeSet`存入自定义对象时，自定义类没有实现`Comparable`接口，或者没有传入`Comparator`较器时，会出现`ClassCastException`异常

Map体系

`Map`与`List`、`Set`接口不同，它是由一系列键值对组成的集合，提供了key到Value的映射。同时它也有继承`Collection`。在`Map`中它保证了key与value之间的一一对应关系。也就是说一个key对应一个value，所以它不能存在相同的key值，当然value值可以相同。实现map的有：`HashMap`、`TreeMap`、`ashTable`、`Properties`、`EnumMap`。

从源码可以看到`TreeSet`、`HashSet` 内部就是通过对应的map 来实现的所以他们key的存储特点及原理一致的

HashMap与TreeMap区别：

- 1、`HashMap`通过hashcode对其内容进行快速查找，而`TreeMap`中所有的元素都保持着某种固的顺序，如果你需要得到一个有序的结果你就应该使用`TreeMap`（`HashMap`中元素的排列顺序是不定的）。`HashMap`中元素的排列顺序是不固定的）。
- 2、`HashMap`通过hashcode对其内容进行快速查找，而`TreeMap`中所有的元素都保持着某种固的顺序，如果你需要得到一个有序的结果你就应该使用`TreeMap`（`HashMap`中元素的排列顺序是不定的）。集合框架”提供两种常规的Map实现：`HashMap`和`TreeMap` (`TreeMap`实现`SortedMap`接)。
- 3、在`Map` 中插入、删除和定位元素，`HashMap` 是最好的选择。但如果您要按自然顺序或自定义顺序遍历键，那么`TreeMap`会更好。使用`HashMap`要求添加的键类明确定义了`hashCode()`和 `equals()`的实现。这个`TreeMap`没有调优选项，因为该树总处于平衡状态。

Hashtable与Hashmap区别

- 1、历史原因:Hashtable是基于陈旧的Dictionary类的，`HashMap`是Java 1.2引进的Map接口的个实现。
- 2、同步性:Hashtable是线程安全的（方法上有加`synchronized`关键字），也就是说同步的，

HashMap是线程不安全的，不是同步的。

3、值：只有HashMap可以让你将空值作为一个表的条目的key或value。