Java Collections Framework集合框架

1. 容器：
2. **数组**：存储**同种类型**的数据，数组是**固定长度**的；
3. **StingBuffer或StringBuilder**：把各种类型数据进行转换成**字符串**，但是一旦转换成**字符串**后，不再回复**原来的类型**；
4. **对象**：把**各种类型的数据**进行封装；
5. **集合**：集合是用于**存储对象**的容器（只能存储对象，不可以存储原生数据类型），集合的长度是**可变的，动态变化的**。
6. **集合的由来**：对象用于封装**特有数据**，对象多了就需要集中**存储**，如果对象的个数不确定，就是用**集合对象**进行存储。（**集合本身也是一个对象**）
7. **集合类的特点**：
8. 用于**存储对象（引用）的容器**；
9. **集合的长度是可变的**；
10. 集合中不可以**存储基本数据类型值**。
11. **集合容器因为其中的数据结构不同，有多种具体容器，不断地向上抽取，就形成了集合框架。关于集合的接口或类都存在于java.util包中。**

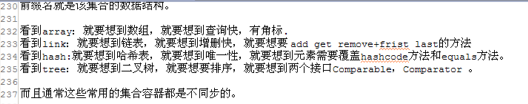
**Java中关于集合这部分主要是四大接口：Collection、List、Set、Map。除此之外，还有一个比较重要的，就是迭代器接口Iterator。还有一些重要的接口，如Enumeration、Comparator等接口。**

**注意：Iterator<E>接口属于java.lang包中，而Iterator<E>属于java.util包中。同理，Comparable<T>接口属于java.lang包中，而Comparator<T>属于java.util包中。注意两对并不是接口继承的关系。**

1. **（1）Java中的集合存在一个根接口：Collection接口，集合框架体系的共性不断的抽取，最终在接口Collection中体现。接口Collection就是Collection层次结构的根接口，它定义了一些最基本的访问方法，让我们能够用统一的方式通过它的货它的子接口来访问数据。**

**（2）Map接口和Collection接口是并列的，Map体系集合是双列集合，而Collection体系集合是单列集合。**

1. **集合的总结：**



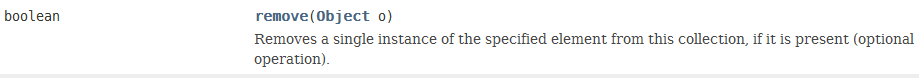
1. **Collection中的主要方法：**
2. **添加方法：add**
3. **add（E e）方法：返回boolean值，成功为true，失败为false。**



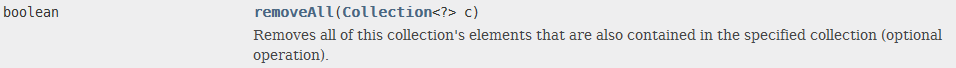
1. **addAll（Collection<?extends E>c）:把一个集合添加到另外一个集合中去，返回boolean值。**



1. **删除和清空方法：（注意区别delete、remove和clear）**
2. **remove（Object o）方法：删除某个集合元素，返回boolean值。**



1. **removeAll（Collection c）方法：删除两个集合的交集，返回boolean值。只要存在交集，且删除此交集，就返回true，只有当两个集合不存在交集时，返回false。即如果此集合元素发生了变化就返回true，否则返回false。**



1. **清空方法：clear（）方法：返回void。**



1. **removeIf（）：新方法。**



1. **包含判断方法：**
2. **contains方法：返回Boolean值。**



1. **containsAll方法：返回Boolean值。**



1. **判断空方法：**

**isEmpty（）方法：**



1. **判断容量方法：size（）方法：获取集合中元素的个数。**

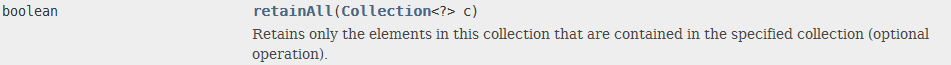


1. **（获取）iterator方法：（返回一个迭代器，用于取出元素所用）**

**iterator** [ɪtə'reɪtə] [想启用英文朗读功能吗？请先安装flash插件！](http://www.adobe.com/shockwave/download/download.cgi?P1_Prod_Version=ShockwaveFlash)n. 迭代器



1. **retainAll方法：（相当于取交集）retain保留，保持：保留下两个集合中重叠的元素，当且仅当此集合中的元素发生变化的时候，才返回true，否则返回false。（注意：removeAll是删除交集）**



1. **toArray方法：将集合转换成数组，具体介绍见Java中的Arrays类介绍。**
2. **返回Object数组：**



1. **返回T数组：**

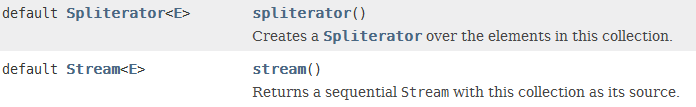


1. **equals方法和hashcode方法：**



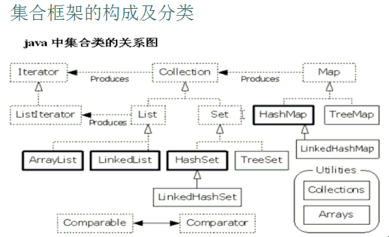


1. **其他方法:**

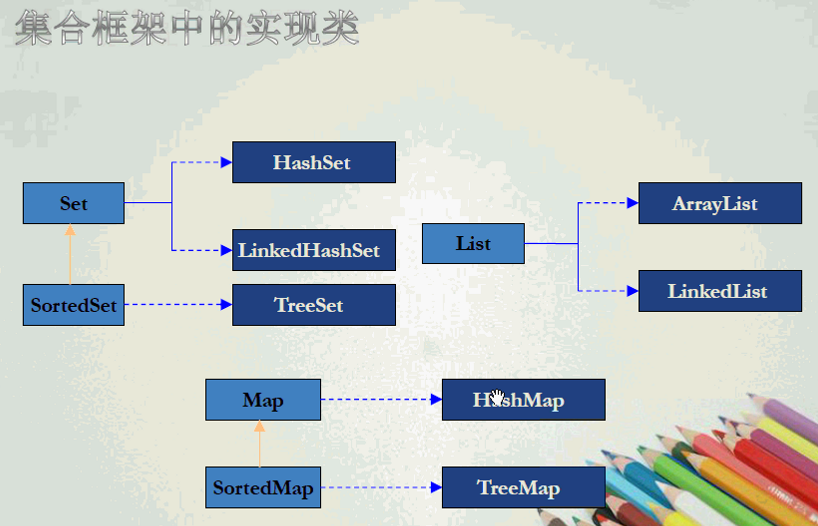


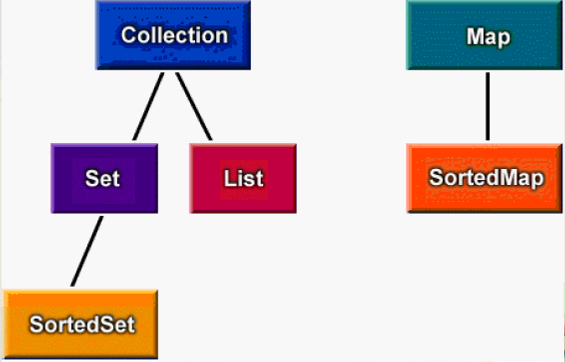


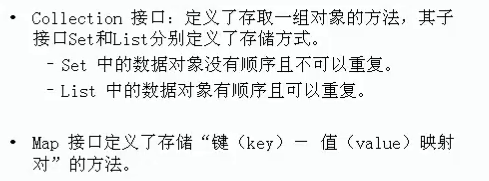
1. **集合框架结构(Collections Framework):**
2. **第一种：**



1. **第二种:**







1. **Collection：接口是集合体系中的根接口。Collection接口没有实现类。Collection接口下面有两个常见的子接口 List（列表）和Set（集）。List和Set有实现类，它们的实现类都是单列集合。注意：Map不是单列集合。**

**（1）List 是有序的，存入和取出的顺序一致，元素都有索引（角标），元素可以重复。**

**（2）Set 是无序的，（有可能有序），元素不能重复，自动对元素进行排序存储。**

**（3） Map为映射，也是一个接口。**

1. **集合里面装的都是对象的引用。集合中存放的只能是对象的引用，不是对象本身，也不能存放原生数据类型，需要使用原生数据类型，其包装类能放到集合当中，集合中存放的都是Object类型的引用，因此取出来的也是Object类型的引用，必须使用强制类型转换（向下转型）将其转换成真正的类型即放置进去的类型。**
2. 既然集合中接收的是对象，但是如果直接传个**基本数据类型**可以吗？

**Object 可以接收任意对象，也可以接收基本数据类型**，因为从JDK1.5开始，基本数据类型便有了自动装箱和自动拆箱功能，故把基本数据类型传进去，自动转换成相应对象，然后利用Object类引用进行接收。

Object obj = 88；//自动装箱

int x = obj + 33；//自动拆箱

1. 每一个类都是继承于Object类的，但是对于**数组**来说，不是继承于**Object数组**的，如Integer[]不是继承于Object[],应该是Integer数组的每个元素（Integer对象）是继承Object类的。
2. 集合使用的一些技巧：
3. 需要**元素唯一**则**选用Set**，**不需要选择List**；
4. 需要**元素唯一**，且**需要指定排序**则选用**TreeSe**t，**不需要排序**则选用HashSet；
5. 不需要元素唯一，但需要频繁增删则选用**LinkedList**；不需要增删则选用ArrayList；
6. 怎样判断容器的结果和**所属体系**？

看名字即可。

**后缀名就是该集合所属的体系，前缀名就是该集合的数据结构。**

**体系：**

**List ： ArrayList ,LinkedList**

**Set: HashSet,TreeSet**

**数据结构：**

**① 看到array就是数组结构，数组结构查询速度快，有角标；**

**② 看到link就是链表结构，链表结构增删速度快，从而就有了add 、get、remove、+ first或last等方法；**

**③ 看到hash就要想到哈希表，就要想到唯一性，就要想到元素需要覆盖hashCode和equals方法。**

**④ 看到tree，就要想到二叉树，就要想到排序，就要想到两个接口（Comparable和Comparator）。**

**通常情况下，这些常用的集合容器都是不同步的。**