

Java集合学习手册（8）：Java 集合框架

一、概念

1.1 什么是集合？

Java官方的入门文档是这样描述集合的：

Collection(有时候也叫container)是一个简单的对象，它把多个元素组织成一个单元。集合可以用来存储、检索、操作、通信。通常情况下，集合代表了一个自然数据项，比如一组手牌(牌的集合)、邮件文件夹(邮件的集合)、电话目录(姓名到电话的映射)。如果你使用过Java或者其他语言，你应该很熟悉集合。

1.2 什么是集合框架？

Collections Framework是一个用来表示和操作集合的统一的架构。集合的框架包括了：

- Interfaces:这些是表示集合的抽象数据类型，接口允许集合完成操作，独立与其详细的实现。在面向对象的语言中，接口构成了体系架构；
- Implementations:这些是接口的具体实现。本质上，是一些可复用的数据结构；
- Algorithms:这些方法可以对接口的对象进行有用的计算，比如搜索、排序。这些算法是具有多态性的：也就是说，同样的方法可以用在合适的接口的不同实现。本质上，是一些可复用的函数。

除了Java的集合框架，还有一些著名的集合框架的例子：比如C++的STL和Smalltalk的集合架构。从历史上来看，集合框架可能比较复杂，也可能有一些很陡峭的学习曲线。不过我们相信Java的集合框架会突破这样的传统，在这章你就可以自己学会。

1.3 使用集合框架有什么好处？

Java的集合框架提供了一下优点：

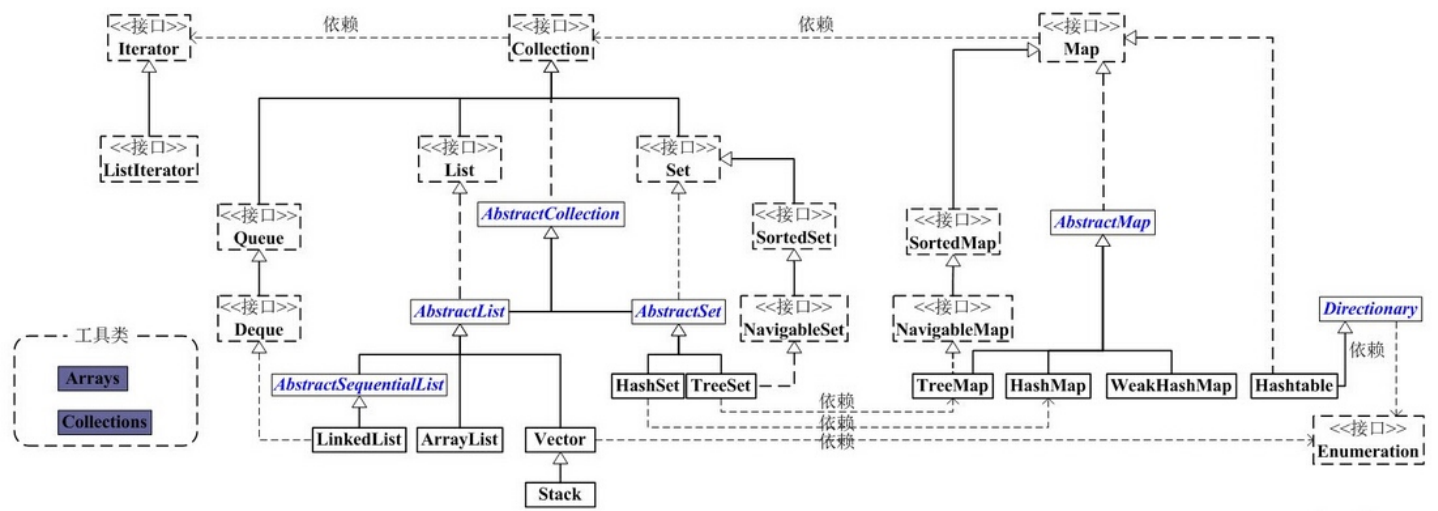
- 减少编程的工作量：通过提供有用的数据结构和算法，集合框架能让你更专注的实现程序的核心功能，而不是去做一个底层的“管道工”。Java框架通过促进无关API的互操作性，使得你不用自己去实现不同API的适配

- 提高程序的速度与质量：集合框架提供了一些有用数据结构和算法的高性能、高质量的实现。每个接口的不同的实现也是可以互换的，所以程序可以通过切换集合来做一些调整。正因为你从实现数据结构的那些苦差事中脱离出来，你才可以有更多的实现去改善你自己程序的性能和质量
- 允许无关APIs的互操作：集合接口是API之间传递集合的一个“方言”，比如我的网络管理API有一个节点名的集合，而GUI工具需要一个列标题的集合，即使是分开实现它们，我们的APIs也可以无缝的接合。
- 省力地学习和使用新API：这是另一个领先的优势，设计者和实现者没必要在每次都重新设计API的时候都“推倒重来”地实现集合，而是直接使用标准的集合接口就好了。
- 促进软件的复用：符合标准集合接口的新数据结构本质上是可以复用的。对于操作这些新数据结构算法也是一样可以复用的。

二、集合框架

Java集合工具包位于Java.util包下，包含了很多常用的数据结构，如数组、链表、栈、队列、集合、哈希表等。学习Java集合框架下大致可以分为如下五个部分：List列表、Set集合、Map映射、迭代器（Iterator、Enumeration）、工具类（Arrays、Collections）。

Java集合类的整体框架如下：



从上图中可以看出，集合类主要分为两大类：Collection和Map。

Collection是List、Set等集合高度抽象出来的接口，它包含了这些集合的基本操作，它主要又分为两大部分：List和Set。

List接口通常表示一个列表（数组、队列、链表、栈等），其中的元素可以重复，常用实现类为ArrayList和LinkedList，另外还有不常用的Vector。

另外，LinkedList还是实现了Queue接口，因此也可以作为队列使用。

Set接口通常表示一个集合，其中的元素不允许重复（通过hashCode和equals函数保证），常用实现类有HashSet和TreeSet，HashSet是通过Map中的HashMap实现的，而TreeSet是通过Map中的TreeMap实现的。另外，TreeSet还实现了SortedSet接口，因此是有序的集合（集合中的元素要实现Comparable接口，并覆写Comparator函数才行）。我们看到，抽象类AbstractCollection、AbstractList和AbstractSet分别实现了Collection、List和Set接口，这就是在Java集合框架中用的很多的适配器设计模式，用这些抽象类去实现接口，在抽象类中实现接口中的若干或全部方法，这样下面的一些类只需直接继承该抽象类，并实现自己需要的方法即可，而不用实现接口中的全部抽象方法。

Map是一个映射接口，其中的每个元素都是一个key-value键值对，同样抽象类AbstractMap通过适配器模式实现了Map接口中的大部分函数，TreeMap、HashMap、WeakHashMap等实现类都通过继承AbstractMap来实现，另外，不常用的HashTable直接实现了Map接口，它和Vector都是JDK1.0就引入的集合类。

Iterator是遍历集合的迭代器（不能遍历Map，只用来遍历Collection），Collection的实现类都实现了iterator()函数，它返回一个Iterator对象，用来遍历集合，ListIterator则专门用来遍历List。而Enumeration则是JDK1.0时引入的，作用与Iterator相同，但它的功能比Iterator要少，它只能在Hashtable、Vector和Stack中使用。

Arrays和Collections是用来操作数组、集合的两个工具类，例如在ArrayList和Vector中大量调用了Arrays.copyOf()方法，而Collections中有很多静态方法可以返回各集合类的synchronized版本，即线程安全的版本，当然了，如果要用线程安全的结合类，首选Concurrent并发包下的对应的集合类。