# 集合

集合的作用：

Java是面向对象的语言，当数据比较多时，比如一个人有年龄，性别等等，这时候我们就用一个对象“人”去储存这些信息，这些信息就成为对象的属性。

那么当对象多了的时候呢？这时候就有了集合。也就是说集合时用来储存对象的。

集合和数组的比较：

数组虽然也可以储存对象，但是它的长度时固定的。集合的长度是可以变化的。

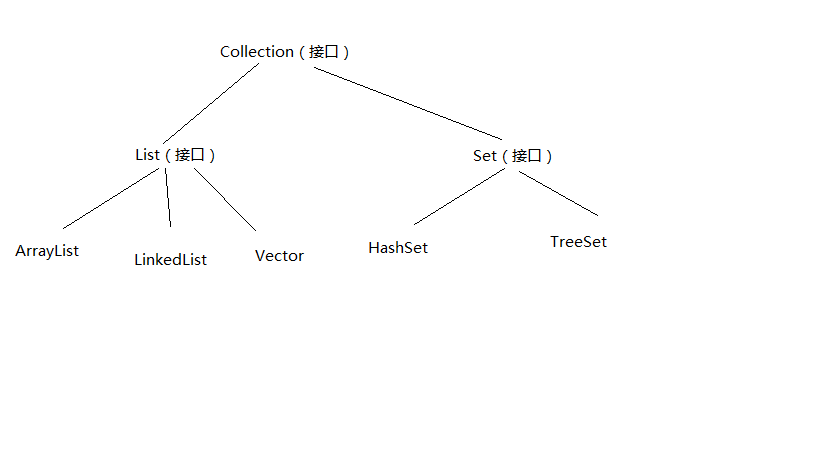
数组存储的必须是一类对像，而集合可以储存任何对象。

数组可以储存基本数据类型，但是集合不可以。

集合框架：

集合就是对象的容器，java提供了许多集合，那么为什么要有这么多的不同的集合呢？因为每一种集合对数据的存储方式是不一样的，这个存储方式就叫做数据结构。不同的数据结构适应不同的需要（主要是对数据不同操作的速度上的不同侧重）。

一，没有Key（单纯只储存对象）的集合框架



对于共性方法的几点说明：

1，集合中存储的都是对象的引用，即内存地址，而非对象本身

2，集合中存储的对象类型是Object，即他能存储所有类型的对象

**迭代器：Iterator**

什么是迭代器？ 其实就是集合的取出元素的方式。

集合是用来存取对象的。所以有个共同的方法add（），只是各自的具体实现方式不太一样。然后那就应该有一个共同的取出方法了，但是，取出是比较复杂的，不是只取就可以了，需要判断里面还有没有元素，还能不能取出。这个时候一个方法就解决不了问题了，就需要一个对象取封装这些方法。

因此这时候，就给集合内部建了一个内部类，内部类可以很方便的去操作这个类的元素。而每一个容器即集合的数据结构是不同的，所以具体的取出方式也是不同的。但是是有共同点：即判断能不能取和取出。这时候就把这些共性抽取。

抽取出一个规则，用接口Iterator描述。那么如果获取集合的内部类，从而获取它取出的方法呢？就让所有集合都向外暴露一个方法iterator（），规定这个方法就是向外提供内部对象的。

从迭代器的实现也能够看出，内部类的作用：即能够很方便的操作类的内部元素。

Collection

|--List:元素是有序的，元素可以重复。因为该集合体系有索引。

|--Set:元素是无需的，元素不可以重复。

List：

凡是能够操作角标的方法都是该提体系特有的方法。