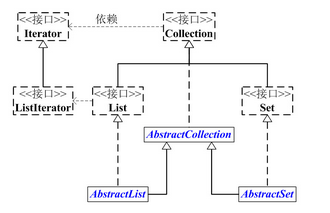
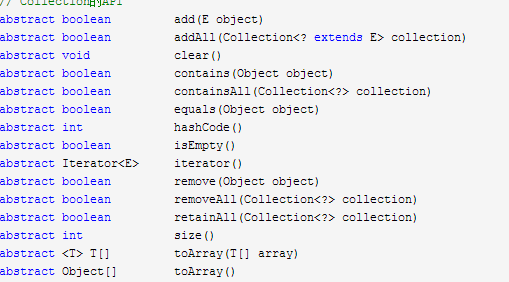
## 1 java集合都有哪些，以及常用到的方法

<http://www.cnblogs.com/skywang12345/p/3308513.html>



Collection是一个接口，主要有两个分支：List和Set；List是有序的队列，以线性方式存储，可以有重复的元素，而Set是数学概念中的集合，Set中没有重复元素。

Collection接口是所有子类的高度抽象的接口 主要的API：



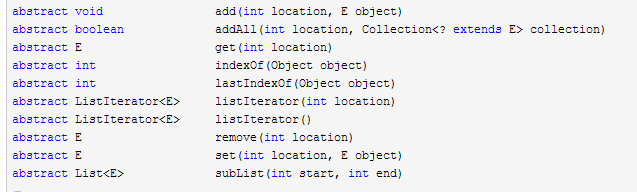
1. List简介

相比于Colleciton，List新增的API有：

add（） 和 set（） 区别：

add是将指定元素添加到此列表的尾部

[set](mk:@MSITStore:E:\work\cisco\Hackathon\JDK_API_1_6_zh_CN\JDK_API_1_6_zh_CN.CHM::/java/util/ArrayList.html#set(int, E))(int index, [E](mk:@MSITStore:E:\work\cisco\Hackathon\JDK_API_1_6_zh_CN\JDK_API_1_6_zh_CN.CHM::/java/util/ArrayList.html) element) 用指定的元素替代此列表中指定位置上的元素。

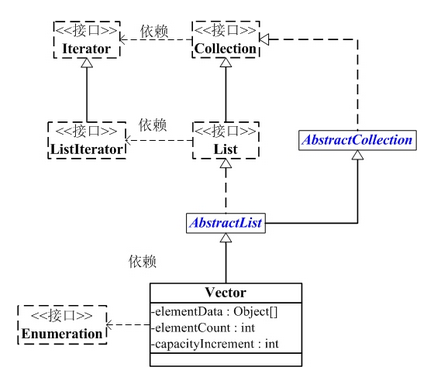


List接口主要实现类包括：

* ArrayList() : 代表长度可以改变得数组。可以对元素进行**随机的访问**，向**ArrayList()中插入与删除元素的速度慢**。
* LinkedList(): 在实现中采用链表数据结构。**插入和删除速度快，访问速度慢**。

当操作是在一列数据的后面添加数据而不是在前面或中间,并且需要随机地访问其中的元素时,使用ArrayList会提供比较好的性能；当你的操作是在一列数据的前面或中间添加或删除数据,并且按照顺序访问其中的元素时,就应该使用LinkedList了。

ArrayList()和Vector()的区别：



基本相同，但是ArrayList()是线程安全的，类似StringBuffer 是线程安全的HashMap是非线程安全的，HashTable是线程安全的

对于List的随机访问来说，就是只随机来检索位于特定位置的元素。 List 的 get(int index) 方法放回集合中由参数index指定的索引位置的对象，下标从“0” 开始。最基本的两种检索集合中的所有对象的方法：

1. set简介

set的api和collection完全一样。

* HashSet： HashSet类按照哈希算法来存取集合中的对象，存取速度比较快。
* TreeSet ：TreeSet类实现了SortedSet接口，能够对集合中的对象进行排序。
* HashSet：哈希表是通过使用称为散列法的机制来存储信息的，元素并没有以某种特定顺序来存放；
* LinkedHashSet：以元素插入的顺序来维护集合的链接表，允许以插入的顺序在集合中迭代；
* TreeSet：提供一个使用树结构存储Set接口的实现，对象以升序顺序存储，访问和遍历的时间很快。
* Iterator 的定义

**Iterator与ListIterator**

是一个接口，它是集合的迭代器。集合可以通过Iterator去遍历集合中的元素。Iterator提供的API接口，包括：是否存在下一个元素、获取下一个元素、删除当前元素。  
注意：Iterator遍历Collection时，是fail-fast机制的。即，当某一个线程A通过iterator去遍历某集合的过程中，若该集合的内容被其他线程所改变了；那么线程A访问集合时，就会抛出ConcurrentModificationException异常，产生fail-fast事件。关于fail-fast的详细内容，我们会在后面专门进行说明。TODO

// Iterator的APIabstract boolean hasNext() abstract E next() abstract void remove()

ListIterator是一个继承于Iterator的接口，它是队列迭代器。专门用于便利List，能提供向前/向后遍历。相比于Iterator，它新增了添加、是否存在上一个元素、获取上一个元素等等API接口。

// ListIterator的API

// 继承于Iterator的接口

abstract boolean hasNext()

abstract E next()

abstract void remove()

// 新增API接口

abstract void add(E object)

abstract boolean hasPrevious()

abstract int nextIndex()

abstract E previous()

abstract int previousIndex()

abstract void set(E object)

1. Map

遍历Map的效率较高的方式：

map.entrySet()是把HashMap类型的数据转换成集合类型  
map.entrySet().iterator()是去获得这个集合的迭代器，保存在iter里面。。迭代器这么用：

**for**(Map.Entry<String, String> entry : map.entrySet()){

System.***out***.println(entry.getKey());

System.***out***.println(entry.getValue());

}

HashMap 在get()的时候执行速度较快，因为HashMap使用了特殊的值，称为散列码（hash code）,来取代键的画满搜索，hash code是相对唯一的，用以代表对象的int值，是通过基类Object的hashCode()方法产生的。

TreeMap 特点类似于TreeSet得到的结果是经过排序的。

## 2 多线程

1. 实现多线程的两种方式：

<http://blog.csdn.net/touch_2011/article/details/6891026>

注意这两种方式的区别：

通过实现Runnable接口创建线程其实并不是真正的线程，因为Runnable接口只是一个普通的接口，只有一个空的run方法。创建一个Runnable对象：TreadDemo1 tt=new TreadDemo1();必须把这个对象附着在一个线程里运行，当把这个对象附着在多个线程里执行时，多个线程是共享tt对象中的资源变量（private int countDown = 10; ），多个线程执行同一个run方法。通过继承Thread创建线程是真正创建了一个线程，每次执行new ThreadDemo2().start(); 都运行一个具有独立资源和run方法的线程，并不会共享资源变量  
如果按照如下方式运行：

1. TreadDemo1 tt=**new** TreadDemo1();
2. **new** Thread(tt).start();
3. TreadDemo1 tt2=**new** TreadDemo1();
4. **new** Thread(tt2).start();
5. System.out.println("火箭发射前倒计时：");

则运行结果和继承Thread类的方法运行结果一样，达不到共享数据的目的了。

1. 线程安全与不安全

<http://blog.csdn.net/xiao__gui/article/details/8934832>

ArrayList和Vector有什么区别？HashMap和HashTable有什么区别？StringBuilder和StringBuffer有什么区别？这些都是Java面试中常见的基础问题。面对这样的问题，回答是：ArrayList是非线程安全的，Vector是线程安全的；HashMap是非线程安全的，HashTable是线程安全的；StringBuilder是非线程安全的，StringBuffer是线程安全的。因为这是昨晚刚背的《Java面试题大全》上面写的。此时如果继续问：什么是线程安全？线程安全和非线程安全有什么区别？分别在什么情况下使用？这样一连串的问题，一口老血就喷出来了…

1. 多线程问题

线程正确的停止方式是设置一个voliate boolean keeprunning = true;声明这个后就可以在其他线程中调用这个变量，改变这个变量的状态值，然后这样就可以，停止该线程。

当多个线程同时共享访问同一数据时，每个线程都尝试操作该数据，从而导致数据被破坏，这种现象称为争用条件。

线程的交互：互斥与同步

互斥：同一时间只有一个线程对数据进行操作。

同步：线程通知另一个线程。

通过使用 synchronized()来声明一段程序，表示是实现同步的，然后lockObj是锁，只有获得这个锁的对象才能自行同步块中的程序，如果条件不满足的时候，线程会阻挡，并且最后会唤醒所有等待的线程。线程会阻挡，并且最后会唤醒所有等待的线程。

Object locObj = new Object()



<http://zangweiren.blog.51cto.com/412366/94386>

1. 线程的状态在java5.0以上版本中，线程状态分为：

new（新建状态，初始化状态）：在调用new命令后，start之前

runnable(就绪状态)：start（）方法后，在jvm上是运行的，正在操作系统分为给它相应的运行资源。

blocked（阻塞状态）：