1、数据结构，研究学习对比几个知名集合框架：谷歌Guava (java.util.Collections)，sun (jdk中的java.util.Collections)，以及一个实时的框架Javolution，三个集合框架的对比。  
2、源码，3种集合框架的常用数据结构的用法比较，测试查10万数据的速度，随机查询速度，内存占用的大小不同。

javolution则是比较知名的一个解决Java实时性的方案

Java有实时性规范

实时性，集合框架，围绕它的性能，案例，最优用法。

有没比Hash方式更快的 字符串查找的数据结构？

堆排序是操作数组么？为什么排序中，堆排序会比较重要，它比选择排序好在哪里？？

http://blog.csdn.net/yutianzuijin/article/details/22876017

https://www.zhihu.com/question/27064078/answer/35156109

Javolution 集合类图

与guava不变集合进行对比

<http://outofmemory.cn/java/guava/Collections/Immutable>

<https://github.com/google/guava/wiki/NewCollectionTypesExplained#bimap>

<https://github.com/google/guava/wiki/ImmutableCollectionsExplained>

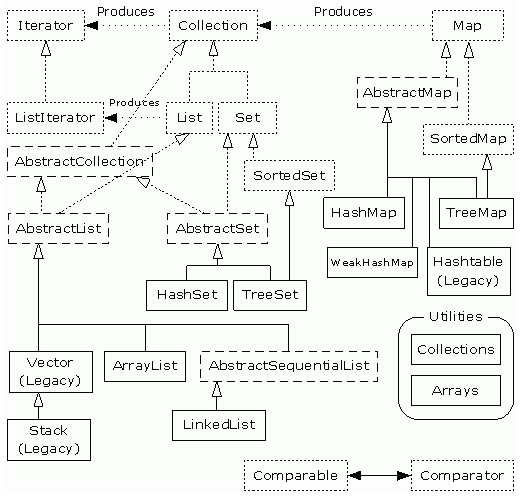
# Java 集合框架与数据结构

<https://dzone.com/articles/biglist-scalable-high>

<http://www.importnew.com/7528.html>

## 1.1 Jdk集合框架分析

### 1.1.1 JDK集合框架类图



首先从数据结构层面来理解整个类图，线性数据结构中两种最重要的两种数据结构就是数组与链表，List列表即是该数据结构的抽象。

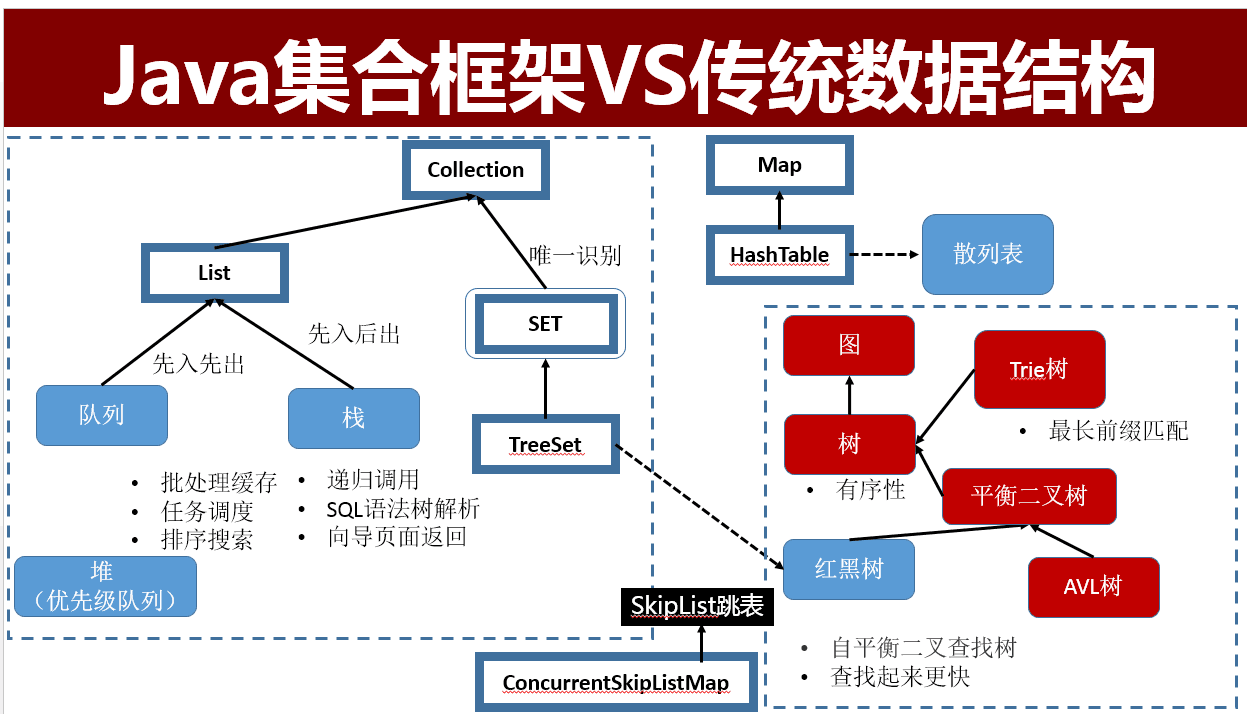
实现List接口的最终工具类ArrayList、Vetor、Stack、LinkedList前三者是数组，后者则是链表。

数组排序算法？

数组在内存里是怎样分配的？连续分配空间。那如何让里面存储的元素也是连续的呢？

ArrayList 与Vetor区别与应用场景？

### 1.1.2 数据结构



为什么树不容易实现高并发

## 1.2 javolution框架分析

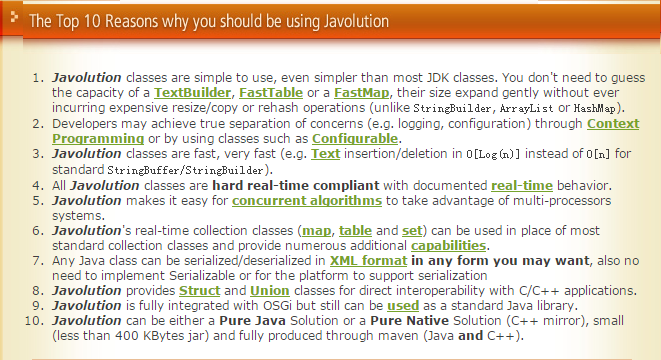
Javolution 提供一个高性的Java集合（collection ）类库和一些实用的工具类。虽然这个类包只提供非常少的几个集合类，但是这些类就能够代替大部分java.util类。javolution可以让你的应用程序更加快速和更实时。

Javolution 主要用在对实时性要求很强的应用中，对集合操作的耗时更加可预测

Javolution高性能实时框架，具体介绍参照官方网址

<http://wjm0729.iteye.com/blog/1561491>

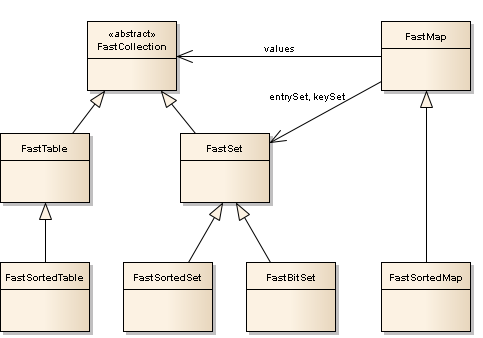
<http://javolution.org/>



实时性是什么？java rts,rtsj

<http://www.oracle.com/technetwork/cn/java/javase/tech/index-jsp-139921-zhs.html>

### 1.2.1 集合类图



### 1.2.1 源码分析

#### 1.2.1.1 FastTable

1.2.1.1 用法与性能对比

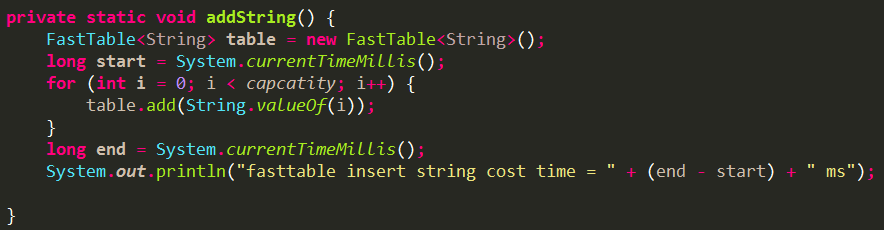
Jvm 分析命令

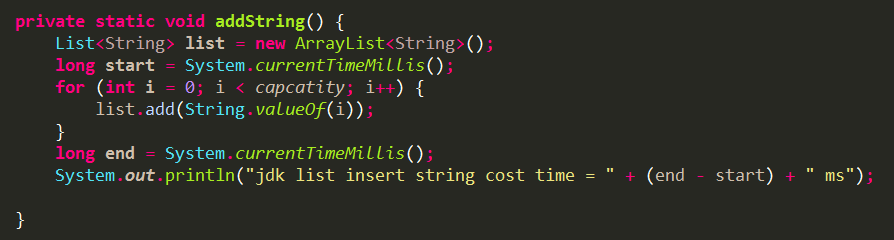
jps

Jstat –gc pid

Jstat –gcutil -pid

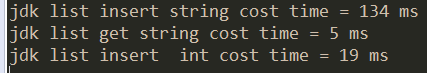
程序代码

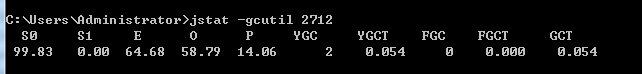




数据量100000，测试结果

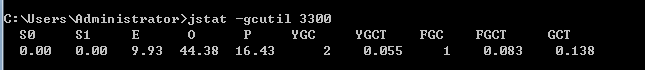
ArrayList：





fastTable:





ArrayList添加删除时间复杂度是o(size)

FastTable 添加删除时间复杂度O(log(size))

实验结果为什么反而慢些？

对比性能、内存占用、内存回收

数据量增加到千万级,10000000

测试结果

ArrayList 直接堆内存溢出



FastTable



官方性能对比



类介绍：

解决的数据结构数组、链表、队列，线程安全，支持并发修改、删除。提供函数式操作。

如何分析源码过程？

功能点：

**fractal tree**

一个类同时满足这么多数据结构有没有增加复杂性？类设计如何满足优雅？

为什么内存占用更少？空间与性能如何同时满足？

#### 1.2.1.2 FastSortedTable

#### 1.2.1.3 FastMap

#### 1.2.1.4 FastSet

#### 1.2.1.5 FastSortedMap

#### 1.2.1.6 FastSortedSet