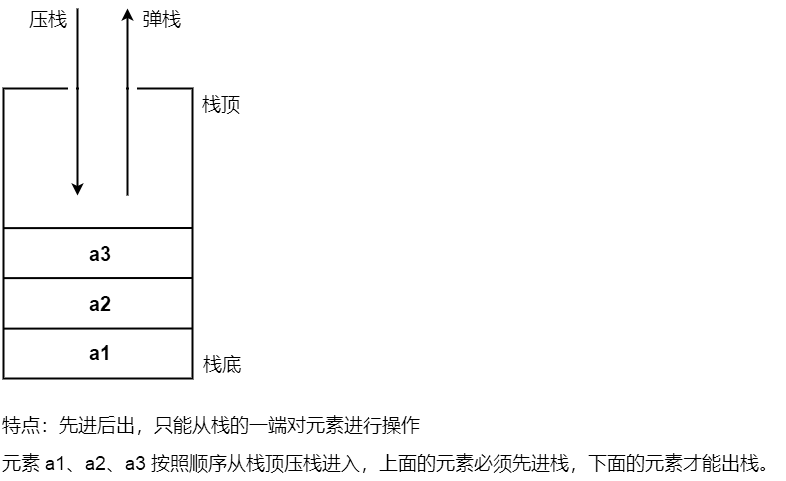
一.数据结构

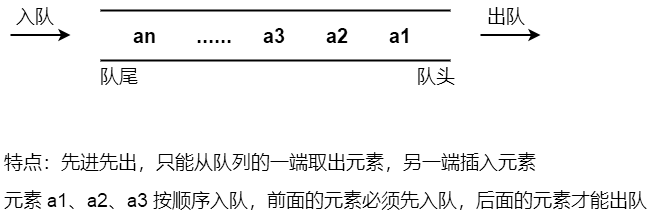
1.栈(Stack)

栈是一种运算受限的线性表，其限制是仅允许在线性表的一端进行插入和删除操作，不允许在其他任何位置进行添加、查找、删除等操作。



2.队列(Queue)

队列也是一种运算受限的线性表，其限制是仅允许在线性表的一端进行插入、仅允许在线性表的另一端进行删除。



3.数组(Array)

数组是有序的元素序列，是在内存中开辟一段连续的空间用来存放数组元素。数组具有以下特点：

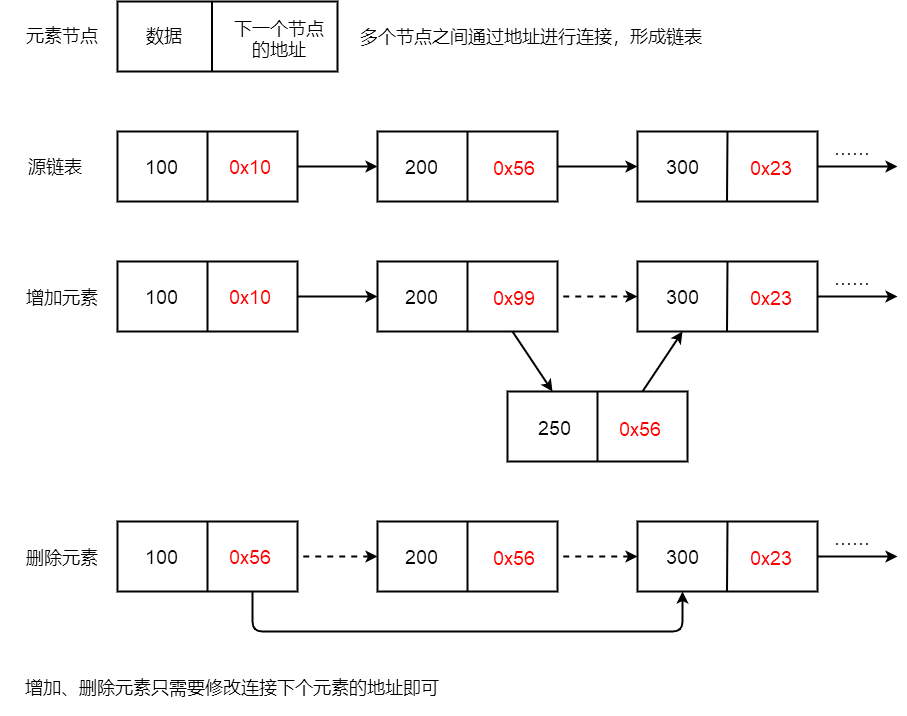
* 查询快：由于数组的地址是连续的，根据数组的首地址、索引，可以快速访问到指定位置的元素。
* 增删慢：由于数组的长度是固定的，增加、删除一个元素，必须先创建一个新的数组，再把源数组的数据复制过来进行增加、删除的操作。



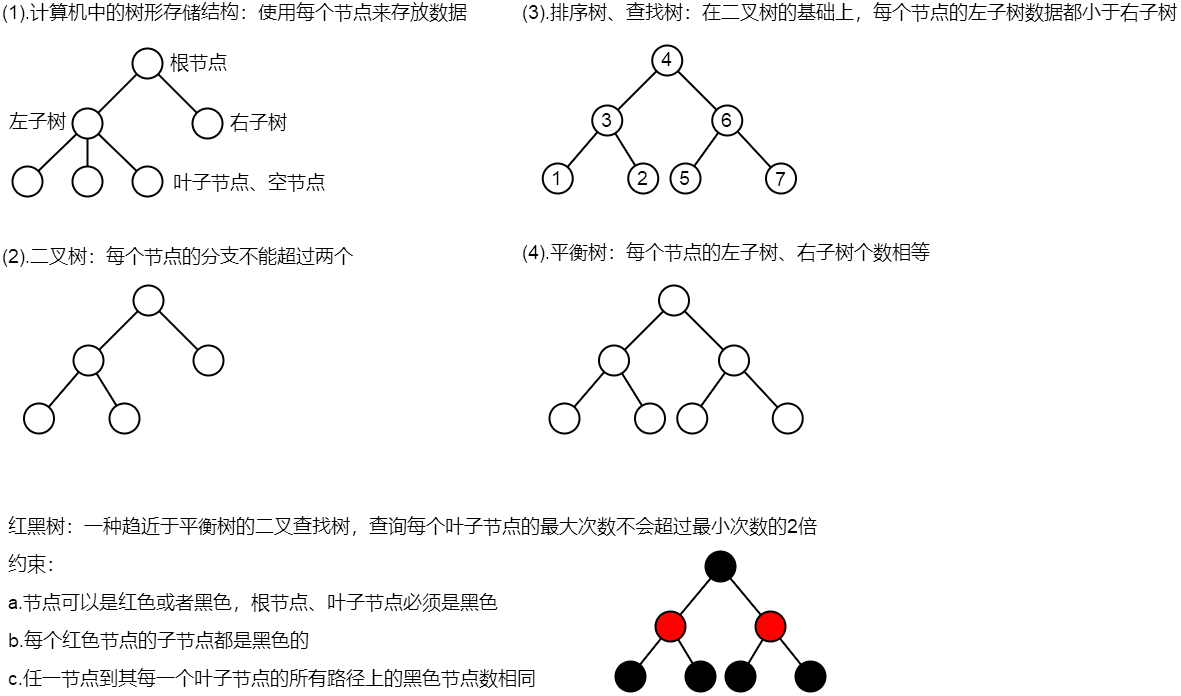
4.链表(Linked List)

链表由一系列节点 node (链表中每一个元素称为一个节点)组成，通过一系列的节点将不连续的内存联系起来，将那种碎片内存进行合理的利用。每个元素节点包括两个部分：存储数据元素的数据域、存储下一个节点地址的指针域。链表具有以下特点：

* 查询慢：由于链表中的地址不是连续的，每次查询元素，都要从头开始查询。
* 增删快：在任意位置增加、删除一个元素，对链表的整体结构没有影响。



5.红黑树



二.List 集合

java.util.List 接口继承自 java.util.Collection 接口，是单列集合的一个重要分支，一般将实现了 List 接口的对象称为 List系集合。List系集合具有以下特点：

* 是一个元素存取有序的集合
* 集合中可以有重复的元素
* 是一个带有索引的集合，通过索引就可以精确的操作集合中的元素

List 作为 Collection 集合的子接口，不但继承了 Collection 接口中的抽象方法，而且还具有一些根据元素索引来操作集合的特有抽象方法。List 接口三个常见的实现类 ArrayList、LinkedList、Vector 会实现 List 接口中的所有抽象方法。(包括从Collection继承过来的抽象方法，见代码 ListUniqueMethod )

1.ArrayList 集合

java.util.ArrayList 集合底层的存储结构是数组结构。元素增删慢、查找快。由于日常开发中多为查询数据、遍历数据，所以 ArrayList 是常用的集合。(ArrayList 类的实现不是同步的，意味着是多线程)

2.LinkedList 集合

java.util.LinkedList 集合底层的存储结构是链表结构。元素增删快、查找慢。LinkedList 类作为 List 接口的实现类，不仅可以使用 List 接口中的方法，自己还拥有一些操作链表首尾元素的特有方法。

3.Vector 集合(了解)

java.util.Vector 集合类是从JDK 1.0开始就有的集合类，但是从JDK 1.2开始就被改进的 ArrayList 集合所替代，所以 Vector 集合与 ArrayList 集合基本一样，区别在于 Vector 集合是线程安全的集合，意味着是单线程的。

三.Set 集合

java.util.Set 接口继承自 java.util.Collection 接口，也是单列集合的一个重要分支。一般将实现了 Set 接口的对象称为 Set系集合。Set系集合具有以下特点：

* 是一个元素存取无序的集合
* 集合中不可以有重复的元素
* 是一个没有索引的集合(不能使用for循环遍历，只能使用迭代器、增强for遍历)

Set 作为 Collection 集合的子接口，它与 Collection 接口中的方法基本一致，并没有对 Collection 接口进行功能上的扩充。Set 集合三个常见的实现类 TreeSet、HashSet、LinkedHashSet 会实现 Set 接口中的所有抽象方法。

1.HashSet 集合

java.util.HashSet 集合的底层其实是一个 java.util.HashMap 支持，即底层存储结构是哈希表。HashSet 类作为 Set 接口的实现类，可以使用 Set 接口中的所有抽象方法。

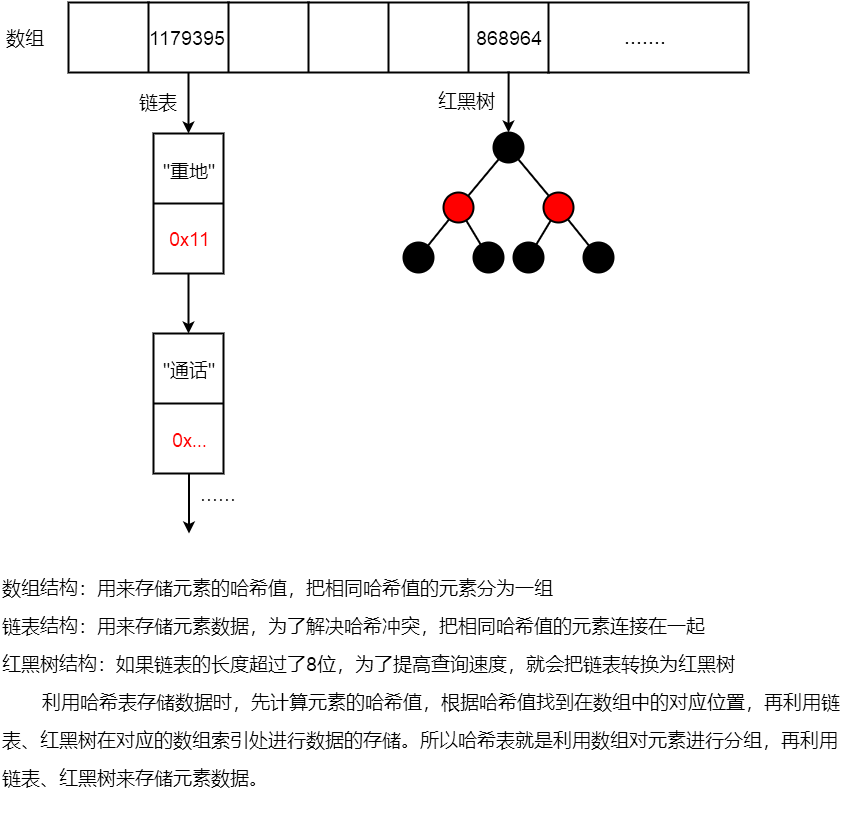
(1).HashSet 集合的用法

(2).HashSet 集合的存储结构——哈希值、哈希表

* 哈希值：是模拟出来的对象的逻辑地址值，不是对象实际存储的物理地址，是一个由系统随机给出十进制整数。使用 Object 类中的 hashCode方法，可以获取对象的哈希值。



* 哈希表：哈希表是一种数据结构，底层采用 "数组+链表+红黑树" 实现，极大地提高了查询速度。



(3).HashSet 集合不存储重复元素的原理



(4).HashSet 集合存储自定义类型元素

HashSet 集合存储自定义类型的元素时，为了保证集合中的对象不重复(属性完全相同的对象我们认为是重复的对象)，自定义的数据类型必须重写 hashCode、equals 方法。(String、Integer类已经重写了hashCode、equals方法)

* 重写 hashCode 方法的原因：由于属性完全相同的对象的哈希值是不一样的，所以要重写 hashCode 方法保证其哈希值一样。
* 重写 equals 方法的原因：见 "重写 Object 类的 equals 方法 "

2.LinkedHashSet 集合

java.util.LinkedHashSet 集合继承了 java.util.HashSet 集合类，所以与 HashSet 集合的原理与使用基本一样。但是 LinkedHashSet 集合的底层存储结构是 "哈希表+链表" ，多出来的链表用来记录元素的存储顺序，因此 LinkedHashSet 集合是存取有序的集合，这一点有别于 HashSet 集合。

3.可变参数

四.Collections 类