

我们检测到你可能使用了 Adblock 或 Adblock Plus，它的部分策略可能会影响到正常功能的使用（如关注）。
你可以设定特殊规则或将知乎加入白名单，以便我们更好地提供服务。（为什么？）

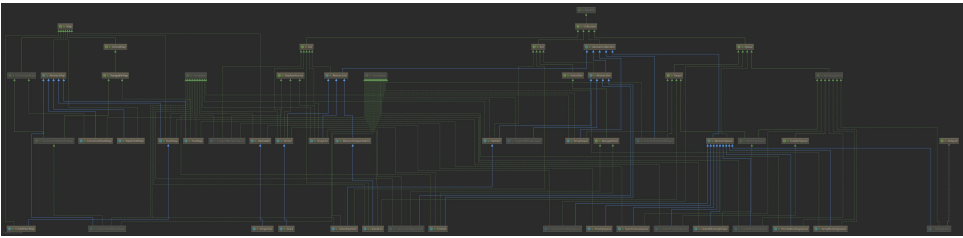
死磕 java集合之终结篇

 **彤哥读源码**
关注公众号“彤哥读源码”，解锁更多源码、基础、架构知识！

18 人赞同了该文章

概览

我们先来看一看java中所有集合的类关系图。

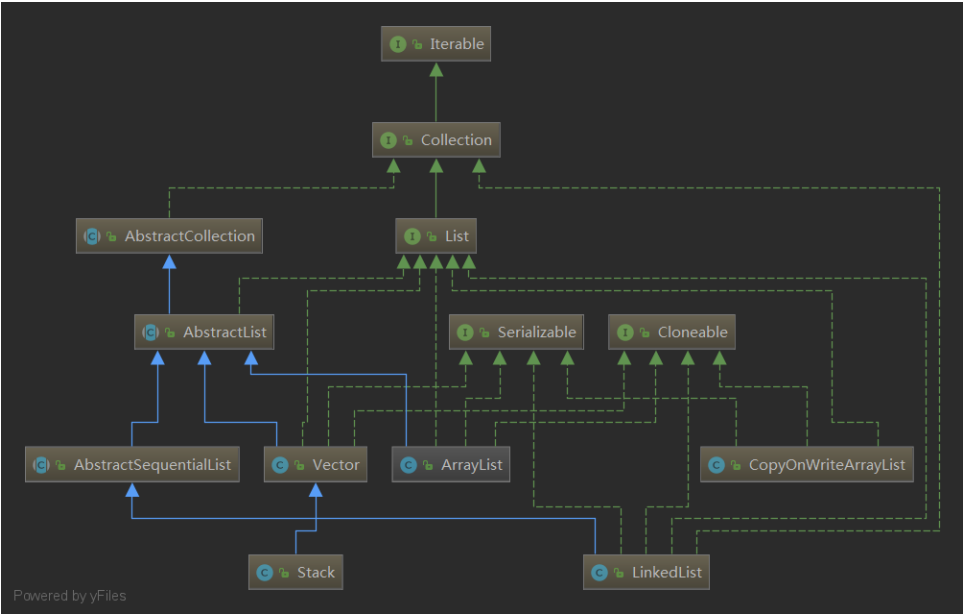


这里的类太多了，请放大看，如果放大还看不清，请再放大看，如果还是看不清，请放弃。

我们下面主要分成五个部分来逐个击破。

List

List中的元素是有序的、可重复的，主要实现方式有动态数组和链表。



java中提供的List的实现主要有ArrayList、LinkedList、CopyOnWriteArrayList，另外还有两个古老的类Vector和Stack。

关于List相关的问题主要有：

▲ 赞同 18 ▼

● 1 条评论

➦ 分享

♥ 喜欢

★ 收藏

📄 申请转载

...

- (3) ArrayList插入、删除、查询元素的时间复杂度各是多少？
- (4) 怎么求两个集合的并集、交集、差集？
- (5) ArrayList是怎么实现序列化和反序列化的？
- (6) 集合的方法toArray()有什么问题？
- (7) 什么是fail-fast？
- (8) LinkedList是单链表还是双链表实现的？
- (9) LinkedList除了作为List还有什么用处？
- (10) LinkedList插入、删除、查询元素的时间复杂度各是多少？
- (11) 什么是随机访问？
- (12) 哪些集合支持随机访问？他们都有哪些共性？
- (13) CopyOnWriteArrayList是怎么保证并发安全的？
- (14) CopyOnWriteArrayList的实现采用了什么思想？
- (15) CopyOnWriteArrayList是不是强一致性的？
- (16) CopyOnWriteArrayList适用于什么样的场景？
- (17) CopyOnWriteArrayList插入、删除、查询元素的时间复杂度各是多少？
- (18) CopyOnWriteArrayList为什么没有size属性？
- (19) 比较古老的集合Vector和Stack有什么缺陷？

关于List的问题大概就这么多，你都能回答上来吗？

点击下面链接可以直接到相应的章节查看：

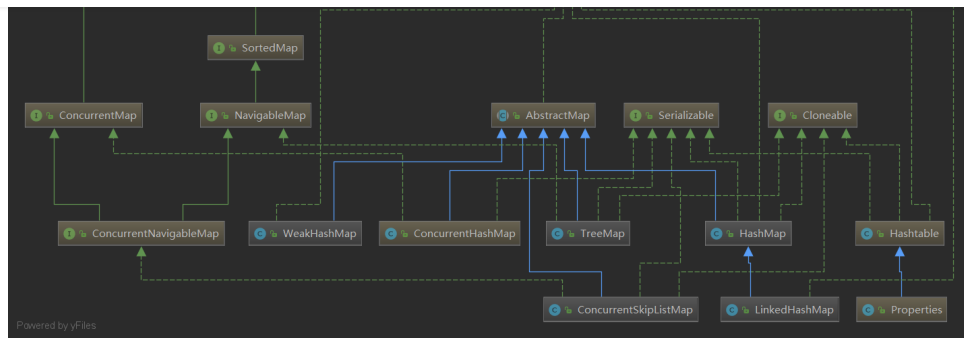
[死磕 Java集合之ArrayList源码分析](#)

[死磕 java集合之LinkedList源码分析](#)

[死磕 java集合之CopyOnWriteArrayList源码分析](#)

Map

Map是一种(key/value)的映射结构，其它语言里可能称作字典（Dictionary），包括java早期也是叫做字典，Map中的元素是一个key只能对应一个value，不能存在重复的key。



java中提供的Map的实现主要有HashMap、LinkedHashMap、WeakHashMap、TreeMap、ConcurrentHashMap、ConcurrentSkipListMap，另外还有两个比较古老的Map实现Hashtable和Properties。

关于Map的问题主要有：

- (1) 什么是散列表？
- (2) 怎么实现一个散列表？
- (3) java中HashMap实现方式的演进？
- (4) HashMap的容量有什么特点？
- (5) HashMap是怎么进行扩容的？
- (6) HashMap中的元素是否是有序的？
- (7) HashMap何时进行树化？何时进行反树化？
- (8) HashMap是怎么进行缩容的？
- (9) HashMap插入、删除、查询元素的时间复杂度各是多少？
- (10) HashMap中的红黑树实现部分可以用其它数据结构代替吗？
- (11) LinkedHashMap是怎么实现的？
- (12) LinkedHashMap是有序的吗？怎么个有序法？
- (13) LinkedHashMap如何实现LRU缓存淘汰策略？
- (14) WeakHashMap使用的数据结构？
- (15) WeakHashMap具有什么特性？
- (16) WeakHashMap通常用来做什么？
- (17) WeakHashMap使用String作为key是需要注意些什么？为什么？
- (18) 什么是弱引用？
- (19) 红黑树具有哪些特性？
- (20) TreeMap就有有序的吗？怎么个有序法？
- (21) TreeMap是否需要扩容？

- (24) 红黑树怎么删除元素?
- (25) 为什么要进行平衡?
- (26) 如何实现红黑树的遍历?
- (27) TreeMap中是怎么遍历的?
- (28) TreeMap插入、删除、查询元素的时间复杂度各是多少?
- (29) HashMap在多线程环境中什么时候会出现问题?
- (30) ConcurrentHashMap的存储结构?
- (31) ConcurrentHashMap是怎么保证并发安全的?
- (32) ConcurrentHashMap是怎么扩容的?
- (33) ConcurrentHashMap的size()方法的实现知多少?
- (34) ConcurrentHashMap是强一致性的吗?
- (35) ConcurrentHashMap不能解决什么问题?
- (36) ConcurrentHashMap中哪些地方运用到分段锁的思想?
- (37) 什么是伪共享? 怎么避免伪共享?
- (38) 什么是跳表?
- (40) ConcurrentSkipList是有序的吗?
- (41) ConcurrentSkipList是如何保证线程安全的?
- (42) ConcurrentSkipList插入、删除、查询元素的时间复杂度各是多少?
- (43) ConcurrentSkipList的索引具有什么特性?
- (44) 为什么Redis选择使用跳表而不是红黑树来实现有序集合?

关于Map的问题大概就这么多, 你都能回答上来吗?

点击下方链接可以直接到相应的章节查看:

[死磕 java集合之HashMap源码分析](#)

[死磕 java集合之LinkedHashMap源码分析](#)

[死磕 java集合之WeakHashMap源码分析](#)

[死磕 java集合之TreeMap源码分析 \(一\)](#)

[死磕 java集合之TreeMap源码分析 \(二\)](#)

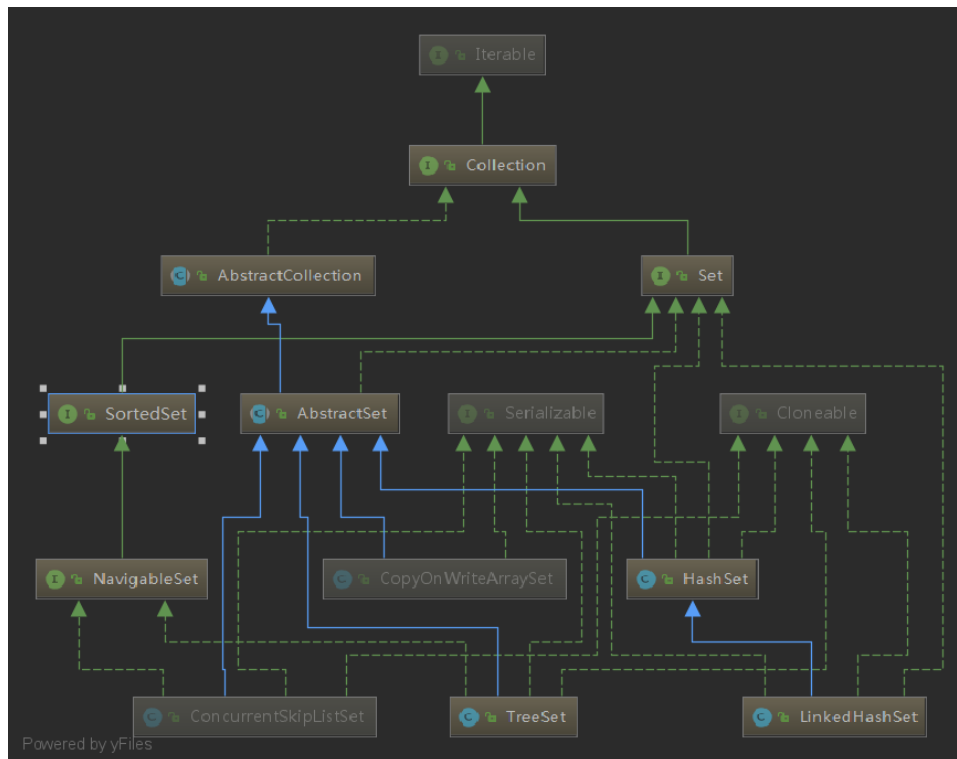
[死磕 java集合之TreeMap源码分析 \(三\)](#)

[死磕 java集合之TreeMap源码分析 \(四\)](#)

[死磕 java集合之ConcurrentHashMap源码分析 \(一\)](#)

Set

java里面的Set对应于数学概念上的集合，里面的元素是不可重复的，通常使用Map或者List来实现。



java中提供的Set的实现主要有HashSet、LinkedHashSet、TreeSet、CopyOnWriteArraySet、ConcurrentSkipSet。

关于Set的问题主要有：

- (1) HashSet怎么保证添加元素不重复？
- (2) HashSet是有序的吗？
- (3) HashSet是否允许null元素？
- (4) Set是否有get()方法？
- (5) LinkedHashSet是有序的吗？怎么个有序法？
- (6) LinkedHashSet支持按元素访问顺序排序吗？
- (8) TreeSet真的是使用TreeMap来存储元素的吗？
- (9) TreeSet是有序的吗？怎么个有序法？
- (10) TreeSet和LinkedHashSet有何不同？
- (11) TreeSet和SortedSet有什么区别和联系？

(14) CopyOnWriteArraySet怎么保证并发安全?

(15) CopyOnWriteArraySet以何种方式保证元素不重复?

(16) 如何比较两个Set中的元素是否完全一致?

(17) ConcurrentSkipListSet的底层是ConcurrentSkipListMap吗?

(18) ConcurrentSkipListSet是有序的吗? 怎么个有序法?

关于Set的问题大概就这么多, 你都能回答上来吗?

点击下方链接可以直接到相应的章节查看:

[死磕 java集合之HashSet源码分析](#)

[死磕 java集合之LinkedHashSet源码分析](#)

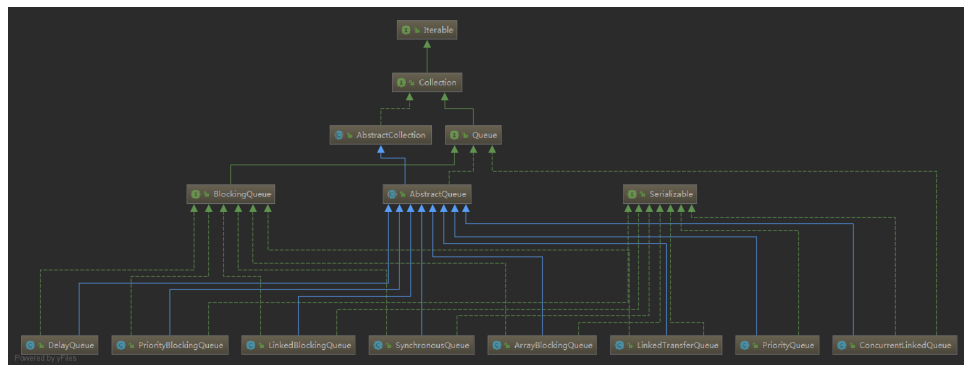
[死磕 java集合之TreeSet源码分析](#)

[死磕 java集合之CopyOnWriteArraySet源码分析](#)

[死磕 java集合之ConcurrentSkipListSet源码分析](#)

Queue

Queue是一种叫做队列的数据结构, 队列是遵循着一定原则的入队出队操作的集合, 一般来说, 入队是在队列尾添加元素, 出队是在队列头删除元素, 但是, 也不一定, 比如优先级队列的原则就稍微有些不同。



java中提供的Queue的实现主要有PriorityQueue、ArrayBlockingQueue、LinkedBlockingQueue、SynchronousQueue、PriorityBlockingQueue、LinkedTransferQueue、DelayQueue、ConcurrentLinkedQueue。

关于Queue的问题主要有:

(1) 什么是堆? 什么是堆化?

(2) 什么是优先级队列?

(3) PriorityQueue是怎么实现的?

(4) PriorityQueue是有序的吗?

- (7) ArrayBlockingQueue的实现方式?
- (8) ArrayBlockingQueue是否需要扩容?
- (9) ArrayBlockingQueue怎么保证线程安全?
- (9) ArrayBlockingQueue有什么缺点?
- (10) LinkedBlockingQueue的实现方式?
- (11) LinkedBlockingQueue是有界的还是无界的队列?
- (12) LinkedBlockingQueue怎么保证线程安全?
- (13) LinkedBlockingQueue与ArrayBlockingQueue对比?
- (14) SynchronousQueue的实现方式?
- (15) SynchronousQueue真的是无缓冲的吗?
- (16) SynchronousQueue怎么保证线程安全?
- (17) SynchronousQueue的公平模式和非公平模式有什么区别?
- (18) SynchronousQueue在高并发场景下会有什么问题?
- (19) PriorityBlockingQueue的实现方式?
- (20) PriorityBlockingQueue是否需要扩容?
- (21) PriorityBlockingQueue怎么保证线程安全?
- (22) PriorityBlockingQueue为什么不需要notFull条件?
- (23) 什么是双重队列?
- (24) LinkedTransferQueue是怎么实现阻塞队列的?
- (25) LinkedTransferQueue是怎么控制并发安全的?
- (26) LinkedTransferQueue与SynchronousQueue有什么异同?
- (27) ConcurrentLinkedQueue是阻塞队列吗?
- (28) ConcurrentLinkedQueue如何保证并发安全?
- (29) ConcurrentLinkedQueue能用于线程池吗?
- (30) DelayQueue是阻塞队列吗?
- (31) DelayQueue的实现方式?
- (32) DelayQueue主要用于什么场景?

关于Queue的问题大概就这么多，你都能回答上来吗？

点击下方链接可以直接到相应的章节查看：

[死磕 java集合之PriorityQueue源码分析](#)

[死磕 java集合之SynchronousQueue源码分析](#)

[死磕 java集合之PriorityBlockingQueue源码分析](#)

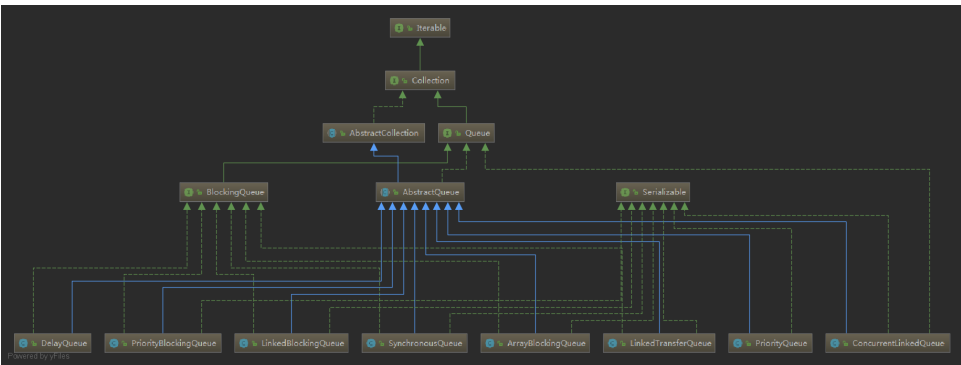
[死磕 java集合之LinkedTransferQueue源码分析](#)

[死磕 java集合之ConcurrentLinkedQueue源码分析](#)

[死磕 java集合之DelayQueue源码分析](#)

Deque

Deque是一种特殊的队列，它的两端都可以进出元素，故而得名双端队列（Double Ended Queue）。



java中提供的Deque的实现主要有ArrayDeque、LinkedBlockingDeque、ConcurrentLinkedDeque、LinkedList。

关于Deque的问题主要有：

- (1) 什么是双端队列？
- (2) ArrayDeque是怎么实现双端队列的？
- (3) ArrayDeque是有界的吗？
- (4) LinkedList与ArrayDeque的对比？
- (5) 双端队列是否可以作为栈使用？
- (6) LinkedBlockingDeque是怎么实现双端队列的？
- (7) LinkedBlockingDeque是怎么保证并发安全的？
- (8) ConcurrentLinkedDeque是怎么实现双端队列的？
- (9) ConcurrentLinkedDeque是怎么保证并发安全的？
- (10) LinkedList是List和Deque的集合体？

关于Deque的问题大概就这么多，你都能回答上来吗？

点击下方链接可以直接到相应的章节查看（LinkedBlockingDeque和ConcurrentLinkedDeque跟相应的Queue的实现方式基本一致，所以笔者没写这两个类的源码分析）：

总结

其实上面的问题很多都具有共性，我觉得以下几个问题在看每个集合类的时候都要掌握清楚：

- (1) 使用的数据结构？
- (2) 添加元素、删除元素的基本逻辑？
- (3) 是否是fail-fast的？
- (4) 是否需要扩容？扩容规则？
- (5) 是否有序？是按插入顺序还是自然顺序还是访问顺序？
- (6) 是否线程安全？
- (7) 使用的锁？
- (8) 优点？缺点？
- (9) 适用的场景？
- (10) 时间复杂度？
- (11) 空间复杂度？
- (12) 还有呢？

彩蛋

到这里整个集合的内容就全部完毕了，其实看了这么多集合的源码之后，笔者发现，基本上所有集合类使用的数据结构都是数组和链表，包括树和跳表也可以看成是链表的一种方式。

对于并发安全的集合，还要再加上相应的锁策略，要不就是重入锁，要不就是CAS+自旋，偶尔也来个synchronized。

所以，掌握集合的源码不算什么，数据结构和锁才是王道。

预告：下一个专题是java并发包，也就是著名的JUC，当然这里是除了并发集合以外的内容，也就是原子类、各种锁、线程池。

欢迎关注我的公众号“彤哥读源码”，查看更多源码系列文章，与彤哥一起畅游源码的海洋。

知乎
首发于
彤哥读源码




彤哥读源码

微信扫描二维码，关注我的公众号

发布于 2019-05-04

Java 源码阅读 JDK源码

文章被以下专栏收录



彤哥读源码

公众号：彤哥读源码

推荐阅读

死磕 java集合之DelayQueue源码分析

问题（1）DelayQueue是阻塞队列吗？（2）DelayQueue的实现方式？（3）DelayQueue主要用于什么场景？简介DelayQueue是java并发包下的延时阻塞队列，常用于实现定时任务。继承体系 从继承...

彤哥读源码 发表于彤哥读源码

Java集合总结【面试题+脑图】，将知识点一网打尽！

前言声明，本文用的是jdk1.8花了一个星期，把Java容器 核心的知识过了一遍，感觉集合已经无所畏惧了！！(哈哈哈....)，现在来总结一下吧~~回顾目录： Collection总览 List集合就这么简单【源...

Java3... 发表于Java3...

Java为于集合

Arraylist 保证线程 LinkedLi 不保证线 构： Arr 数组； Li HChan

1 条评论

切换为时间排序

写下你的评论...

😊

 Smile

03-10

请问，第一个看不清楚的图如何在idea中生成呢

👍 赞