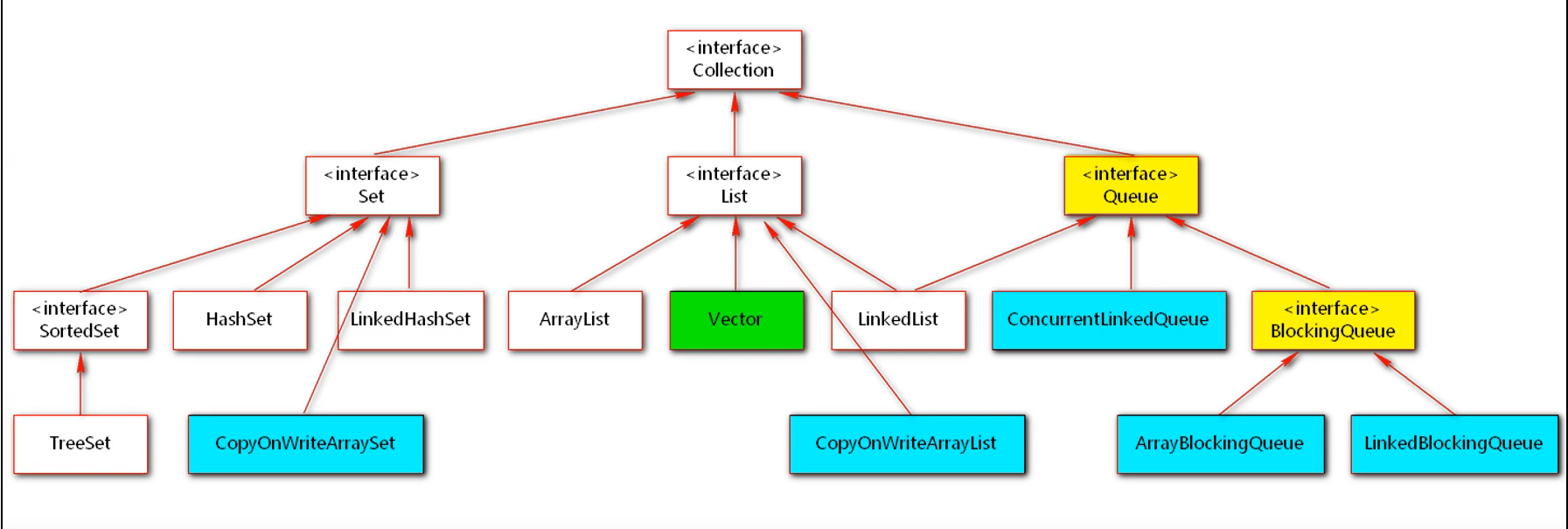
今日任务完成7

线程安全集合

Collection体系集合中，除Vector以外的线程安全集合



蓝色为安全的



CopyOnWriteArraysList

线程安全的ArraysList,加强版的读写分离

写有锁，读无锁，读写之间不阻塞，优于读写锁

写入时，先Copy一个容器副本、再添加新元素，最后替换引用

使用方式与ArrayList无异

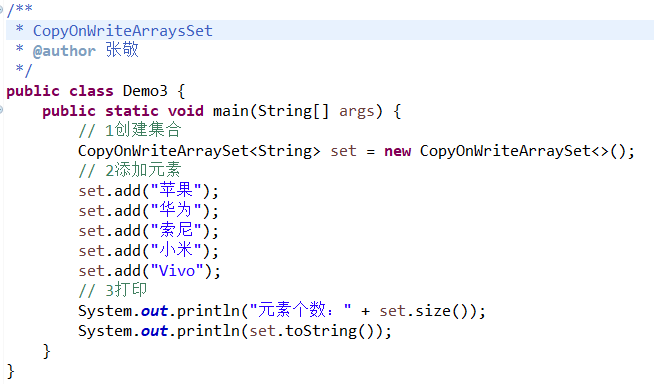


CopyOnWriteArraysSet

线程安全的Set,底层使用CopyOnWriteArraysList实现

唯一不同在于，使用addIfAbsent()添加元素，会遍历数组

如存在元素，则不添加(扔掉副本)



Queue接口

Collection的子接口，表示队列FIFO(先进先出)

常用方法

1. 抛出异常：

boolean add(E e)//顺序添加一个元素（到达上限后，再添加则会抛出异常）

E remove()//获取第一个元素并移除(如果队列没有元素时，则抛出异常)

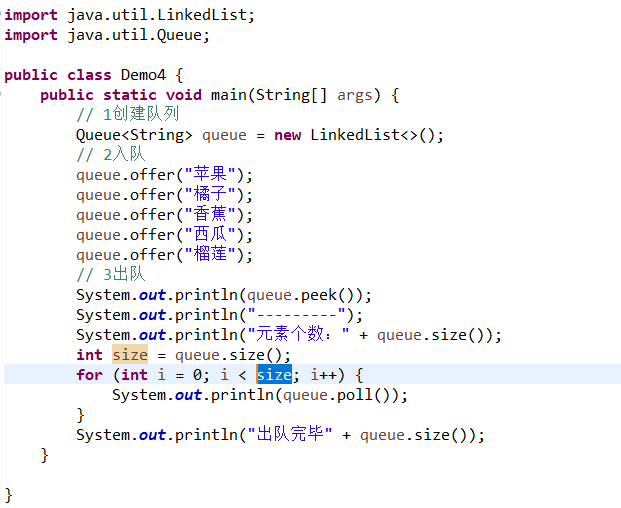
E element()//获得第一个元素但不移除(如果队列没有元素时，则抛出异常)

1. 返回特殊值：推荐使用

boolean offer(E e)//顺序添加一个元素(到达上限后，再添加则返回false)

E Poll()//获得第一个元素并移除(如果队列没有元素时，则返回null)

E peek()//获得第一个元素但不移除(如果队列没有元素时，返回null)



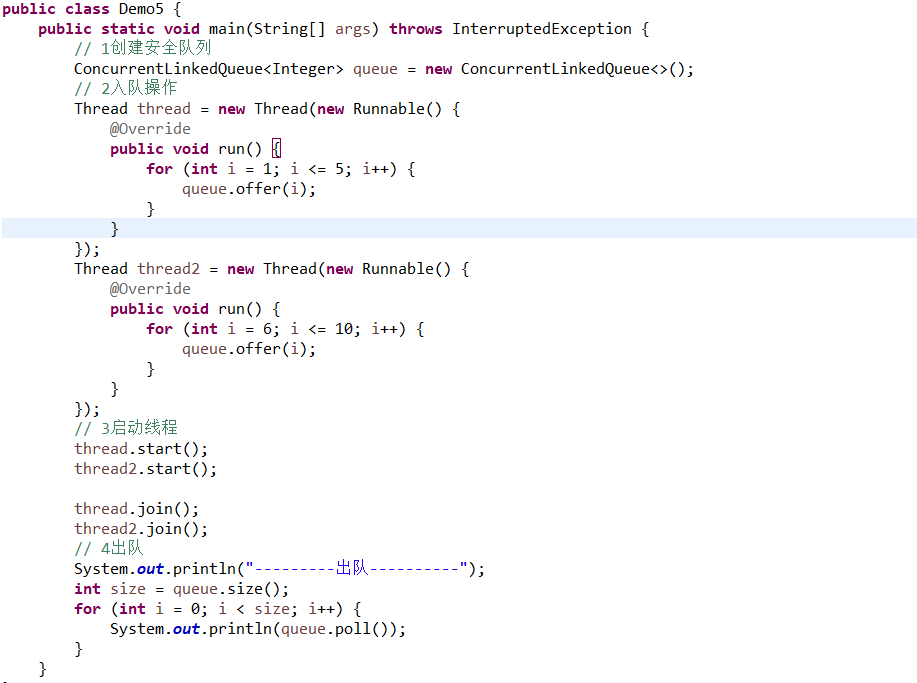
ConcurrentLinkedQueue

线程安全、可高效读写的队列，高并发下性能最好的队列

无锁、CAS比较交换算法，修改的方法包括三个核心参数（V,E,N）

V:要更新的变量 E：预期值 N：新值

只有当V==E时，V=N；否则表示已被更新过，则取消当前操作



BlockingQueue接口(阻塞队列)

方法：

void put(E e)//将指定的元素插入此队列中，如果没有可用空间，则等待

E take()//获取并移除此队列头部元素，每个没有可用元素，则等待

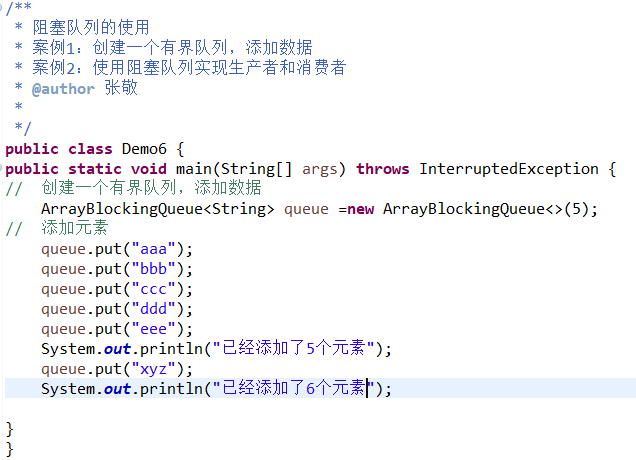
可用于解决生产者、消费者问题

ArrayBlockingQueue:

数组结构实现，有界队列。(手工固定上限)

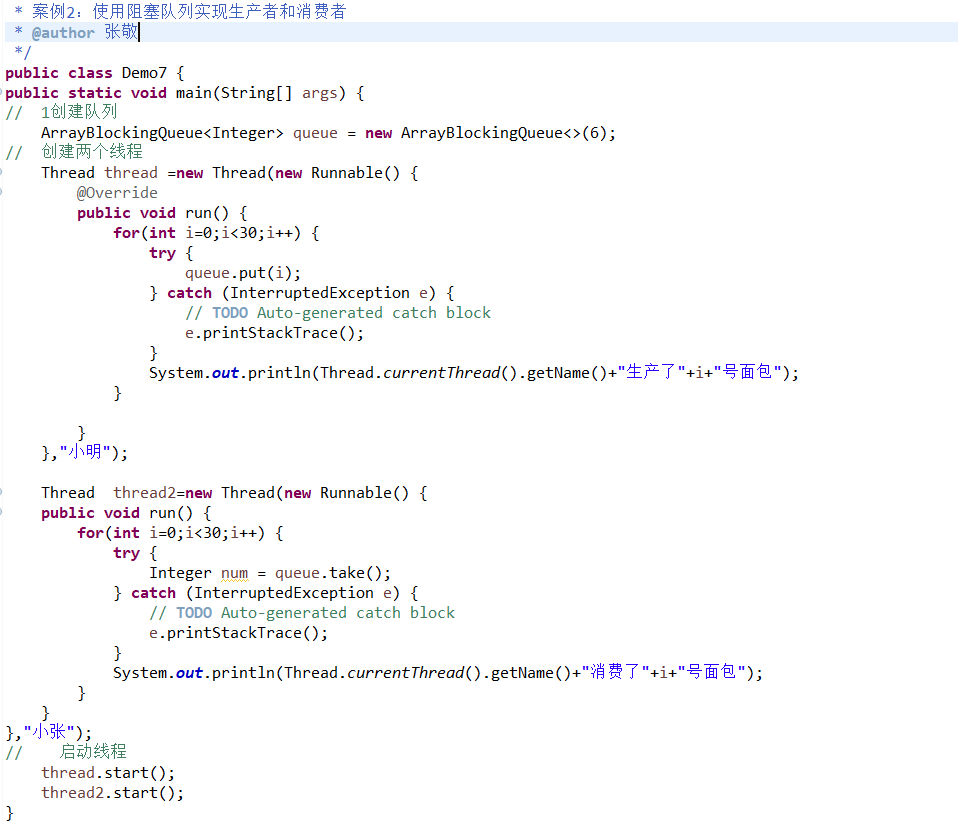
LinkedBlockingQueue:

链表结构实现，有界队列。(默认上限Interger.MAX\_VALUE)



第六个元素被阻塞





ConcurrentHashMap

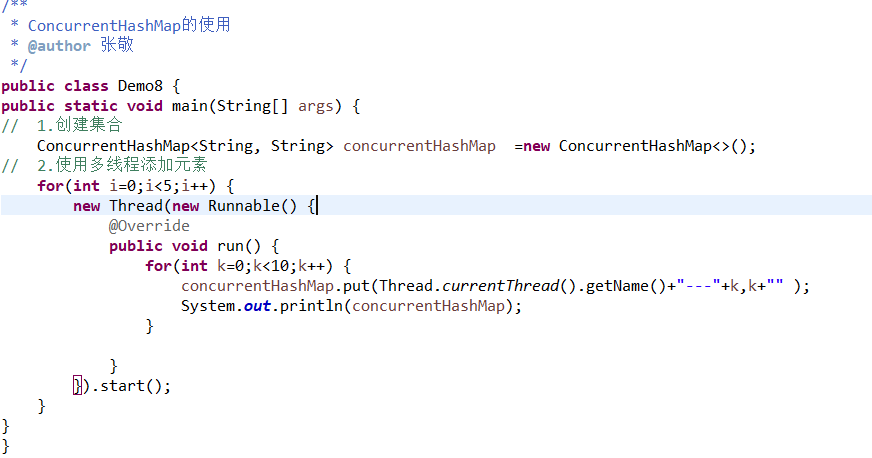
初始容量默认为16段(sagment),使用分段锁设计

不对整个Map加锁，而是为每个Sagment加锁

当多个对象存入同一个Segment时，才需要互斥

最理想状态为16个对象分别存入16个Segment,并行数量16

使用方式与HashMap无异



流

概念：内存与存储设备之间传输数据的通道

流的分类

按方向：输入流：将存储设备中的内容读入到内存中

输出流：将内存中的内容写入到存储设备中

按单位：字节流：以字节为单位，可以读写所有数据

字符流：以字符为单位，只能读写文本数据

按功能：节点流：具有实际传输数据的读写功能

过滤流：在节点流的基础上增强功能