多线程

Wait 和 sleep 的区别

1. wait可以指定时间也可以不指定

sleep必须指定时间

1. 在同步中时，对cpu的执行权和锁的处理不同

Wait：释放执行权，释放锁。

Sleep：释放执行权，不释放锁。

怎么样结束进程？

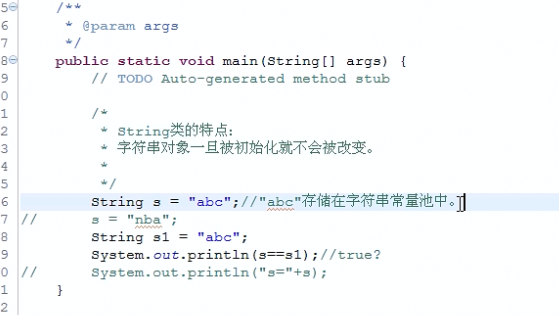
1. stop结束（已过时）
2. run方法结束

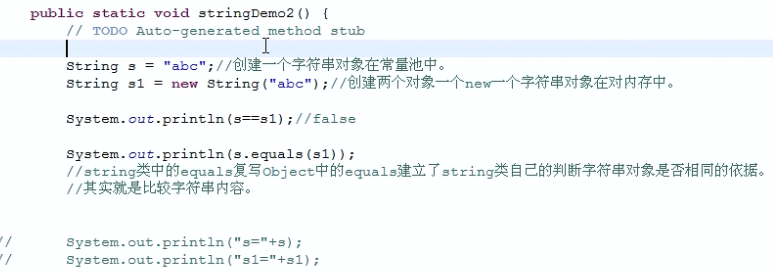
控制循环定义标记结束线程

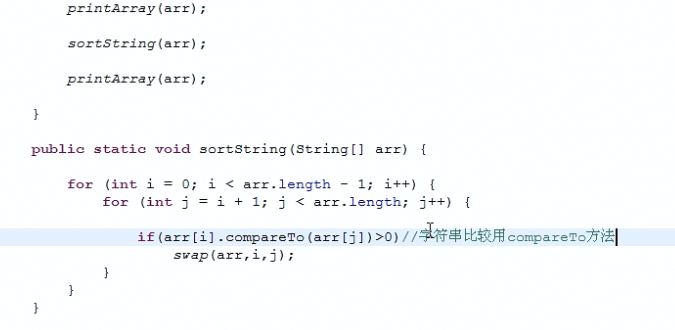
如果线程处于冻结状态，无法读取标记，可以使用interrupt（）方法将线程从冻结状态强制恢复到运行状态中来，但强制动作会发生InterruptedException，记得要处理

String

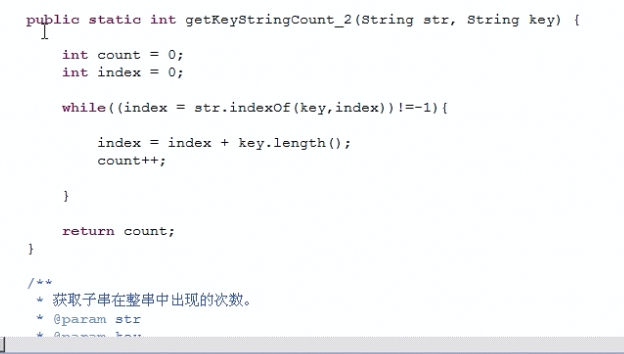
“abc”存储在字符串常量池中。



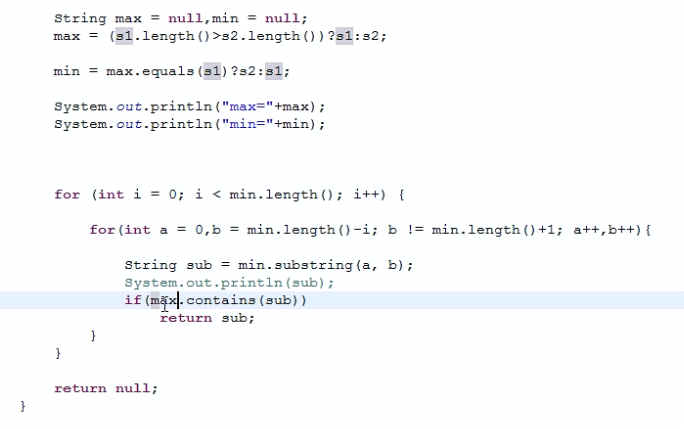


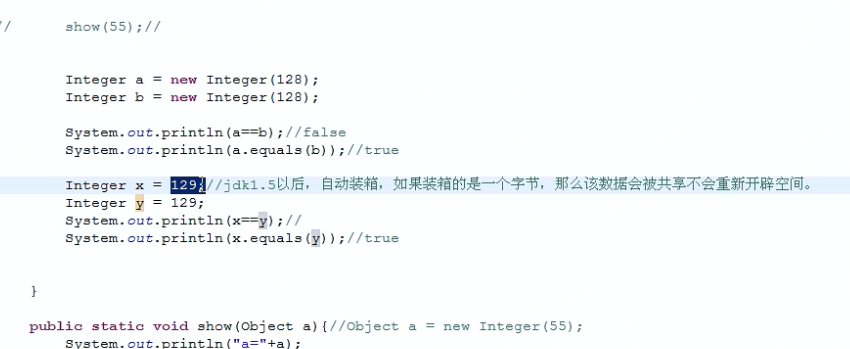


获取字符串中子串的个数



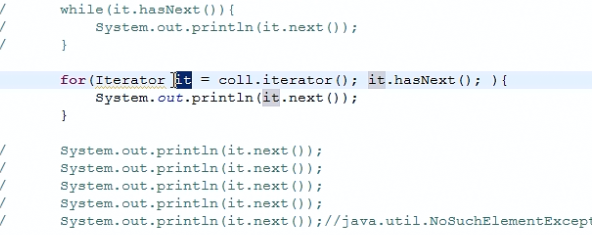
最大相同子串





集合

迭代器遍历



List和set的特点

List：有序的collection（也称序列），存入和取出的顺序一致，元素都有索引（角标）元素可以重复

Set：元素不能重复，无序。有可能有序，

List：

1. Vector:内部是数组数据结构。是同步的。增删，查询都很慢
2. ArrayList: 内部是数组数据结构。是不同步的。替代了Vector。查询的速度很快
3. LinkedList:内部是链表数据结构，是不同步的。增删元素的速度很快

LinkedList：

addFilrst():

addLast();

jdk1.6

offerFrist():

offerLast():

getFirst()://获取但不移除，如果链表为空，抛出NoSuchElementException

getLast():

jdk1.6

peekFirst()://获取但移除，如果链表为空，返回null

peekLast():

removeFirst()://获取但移除，如果链表为空抛出NoSuchElementException

removeLast():

jdk1.6

pollFirst():///获取但移除，如果链表为空返回null

PollLast():

Set: 元素不可以重复

Set接口中的方法和collection一致

HashSet:内部数据结构是哈希表，不同步

TreeSet:

TreeSet覆盖compareTo

Int temp = this.age – p.age;

Return temp == 0? this.name.compareTo(p.name):temp

Map常用的子类

1. HashTable：内部结构是哈希表，是同步的，不允许null作为键和值

Properties：用来存储键值对型的配置文件信息，可以和IO技术相结合

1. HashMap：内部结构是哈希表，不是同步的，允许null作为键和值
2. TreeMap：内部结构是二叉树，不是同步的，可以对Map集合中的键进行排序

泛型：jdk1.5出现的安全机制

好处：

1. 将运行时期的问题ClassCastException转到了编译时期
2. 避免了强制转换的麻烦

<>:什么时候用？当操作的作用数据类型不确定的时候，就使用<>

将要操作的引用数据类型传入即可，

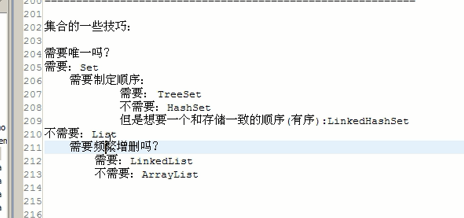
泛型技术是给编译器使用的技术，用于编译时期，确保类型的安全

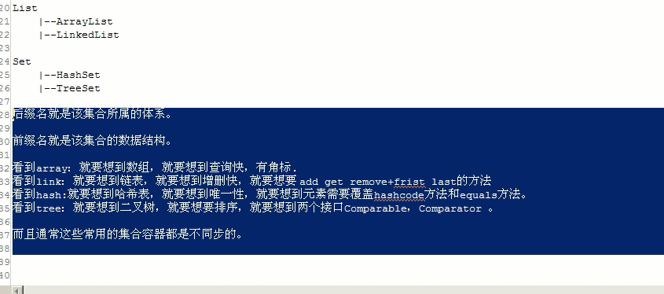
运行时，会将泛型去掉，生成的class文件中是不带泛型的，这个称为泛型的擦除

为了兼容运行的类加载器

泛型的补偿：在运行时，通过获取元素的类型进行转换动作，不用使用者再强制转换







Collections排序

