1. 抽象和接口的区别：
2. 抽象类：在面向对象概念中，所有的对象都是通过类来描述的，但是反过来，并不是所有的类都是用来描绘对象的，如果一个类中没有包含足够的信息来描述一个具体的对象，这个类就是抽象类。
3. 抽象类除了不能实例化对象，类的其他功能依然存在，成员变量，成员方法和构造方法的访问方式和普通类一样。
4. 由于抽象类不能实例化对象，所以抽象类必须被继承才能使用。
5. 接口是指被别人调用的方法或者函数
6. 接口中可以含有变量和方法，但是接口的。变量会被隐式的指定为public static final变量，方法则会被隐式指定为public abstract方法，所以接口中的所有方法都是抽象方法，不能有具体实现。
7. 区别：
   1. 一个类可以继承一个抽象类，但是可以实现多个接口。
   2. 抽象类中的成员变量可以有很多类型，而接口只能是public static final。
   3. 抽象类可以定义普通属性和静态常量属性，而接口只能定义public static final。
   4. 抽象类可以提供成员方法的实现细节，而接口只能存在public abstract。

二. List Set Map的区别：

1. list和set都是实现了collection接口。

2. list：

1. 有get方法
2. 可以允许重复的对象；
3. 可以插入多个null值；
4. 有序容器，保证了每个元素的插入和输出顺序；
5. 常用的实现类有ArrayList（索引的随机访问），LinkedList（list中添加或删除元素）和Vector。

3. set：

1. 没有get方法
2. 不允许重复对象；
3. 只允许一个null值；
4. 无序容器。TreeSet通过Comparator或Comparable维护了一个排序顺序；
5. 常用实现类：HashSet、LinkedHashSet、TreeSet（实现了SortedSet接口）。

4. Map：

1. Map不是collection的子接口或者实现类，它是一个接口；
2. Map的每个Entry都持有两个对象，key和value，key只能有一个，value可以有多个；
3. TreeMap也通过Comparator或者Comparable维护了一个排序顺序；
4. 常见实现类：HashMap、LinkedHashMap、HashTable、TreeMap。

5. 接口Comparable<E>：

此接口强行对实现它的每个类的对象进行整体排序，此排序被称为类的自然排序，类的compareTo方法被称为它的自然比较方法。

三. ArrayList和LinkedList的区别

ArrayList和LinkedList都是List的实现类.

ArrayList：

1. 数组(顺序表的存储结构)。
2. 顺序表的存储地址是连续的,所以查找比较快。
3. 在插入删除的时候，由于需要把其他元素往后或者前移动，比较耗时。
4. 空间花费：要在list列表的结尾预留一定的容量空间
5. 适合索引的随机访问；尾插尾删

LinkedList：

1. 链表。
2. 存储地址不连续，每个存储地址通过指针指向，在查找的时候需要通过指针遍历元素，查找比较慢。
3. 插入删除不需要移动其他元素，比较快。
4. 空间花费：每个元素都需要消耗空间
5. 适合对指定位置进行插入删除

Vector：

1. 采用synchronized，线程安全，性能较低，读读互斥

四. set接口和map接口的关系

1. set实际上就是map，保存的单个元素存储在map的key

2. set不允许数据重复

3. hashSet判断两个对象是否重复，用equals和hashCode

4. 元素想要保存到TreeSet中，要么本身所在的类实现Comparable，要么通过外部传一个比较器

五. HashMap HashTable TreeMap的关系 区别

1. 底层实现：HashMap和HashTable基于哈希表实现，TreeMap基于红黑树实现

2. null值：HashMap的key和value都可以为空，key-1 value-N；

HashTable都不能为空；

TreeMap只允许value为空

3. 线程安全：

HashMap和TreeMap异步处理，不安全，性能高；

HashTable用synchronized，安全，性能低，读读互斥；

六. Java多线程的四种实现方法

1. 继承Thread类
2. 实现Runnable接口
3. 实现Callable接口
4. 使用线程池

七. 如何启动线程

线程只有一种启动的方法，就是调用start方法

Run方法仅仅是启动一个类

八. 线程停止：

设置标记位（正常退出）；

Stop（强行退出）；

Interrupt（中断传输）