|  |  |  |
| --- | --- | --- |
| 类 | 所属接口 | 特性 |
| LinkedList | List  有序的，可重复的 | 本质是双向链表  适合插入频繁的情景 |
| ArrayList | List | 本质是动态数组  线程不安全  适合访问频繁的情景  当容量不足以容纳当前的所有元素时，设置新的容量=（旧容量\*3）/2+1 |
| Vector | List | 本质是矢量队列  线程同步的  实现了枚举接口 |
| Stack | List | 是栈,先进后出 |
| HashSet | Set  不可重复的 | 由HashSet实现  内部用一个HashMap保存内容  线程不安全  无序集合 |
| TreeSet | Set | 通过TreeMap实现  非线程安全  有序集合 |
| TreeMap | Map  键值对，键不可重复 | 本质是红黑树  线程不安全 |
| HashMap | Map | 本质是一个散列表（哈希表）  通过**拉链法**解决哈希冲突  不是线程安全  实现了序列化  默认的初始容量是16，必须是2的幂 |
| WeakHashMap | Map | 键是弱键，不再正常使用时，会被移除 |
| HashTable | Map | 线程同步的  key,value都不可以是null |

## 集合

Hashmap用什么来确保容量必定是2的幂

1.默认构造函数容量为16，加载因子是0.75

2.指定容量大小的构造函数，会调用第三个构造函数

3.指定容量大小和加载因子的构造函数

首先确保指定容量大小小于2^30

其次找出大于指定容量的最小的2的幂并赋值为数组的长度

|  |  |
| --- | --- |
| Arrays(面向数组) | List list=Arrays.asList(array)  Arrays.sort(array)(默认快排) |
| Collections（面向集合） | Collections.sort(list) |

## Stream

|  |  |
| --- | --- |
| Stream | |
| 新建Stream对象 | Stream stream=Arrays.stream(array)  Stream stream=list.stream()  Stream stream=Stream.of(“”,””,””) |
| 排序 | stream. sorted(Comparator.*comparing*(Integer::*parseInt*)).forEach(System.***out***::println) |
| 过滤 | stream.filter(e->e.getSex().equal(“女”)).foreach(System.out::println)  限制stream.filter(e->e.getSex().equal(“女”)).limit(1).foreach(System.out::println)  跳过stream.filter(e->e.getSex().equal(“女”)).skip(1).foreach(System.out::println) |
| 提取 | List<emp>list=Arrays.asList(….)  Stream stream=list.stream()  List<String>list=stream.map(s -> s.getName()).collect(Collectors.*toList*()).forEach(System.***out***::println) |

## 数组

1. 动态初始化

int[] array = new int[3];

1. 静态初始化

int[] array = new int[]{4, 2, 1};

不能同时使用

## ResultSetMetaDate

|  |  |
| --- | --- |
| 方法 | 功能 |
| getColumnCount() | 获取列数 |
| getColumnName(int n) | 获取列名 |

## BeanUtils

|  |  |
| --- | --- |
| 方法 | 功能 |
| setProperty(Object , String ,Object) | 设置类的成员 |
|  |  |

## 内部类

|  |  |  |
| --- | --- | --- |
|  | 静态内部类 | 非静态内部类 |
| 是否需要实例化外部类 | 不需要  outer.inner inner=new inner() | 需要  Outer outer=new outer()  Outer.inner inner=outer.new inner() |
| 是否可以声明静态类 | 可以 | 不可以 |
| 是否可以调用外部类中的成员 | 不全可以，只可以调用外部类的静态方法 | 完全可以 |
| 是否可以定义静态成员 | 可以 | 不可以 |
|  |  |  |
|  |  |  |