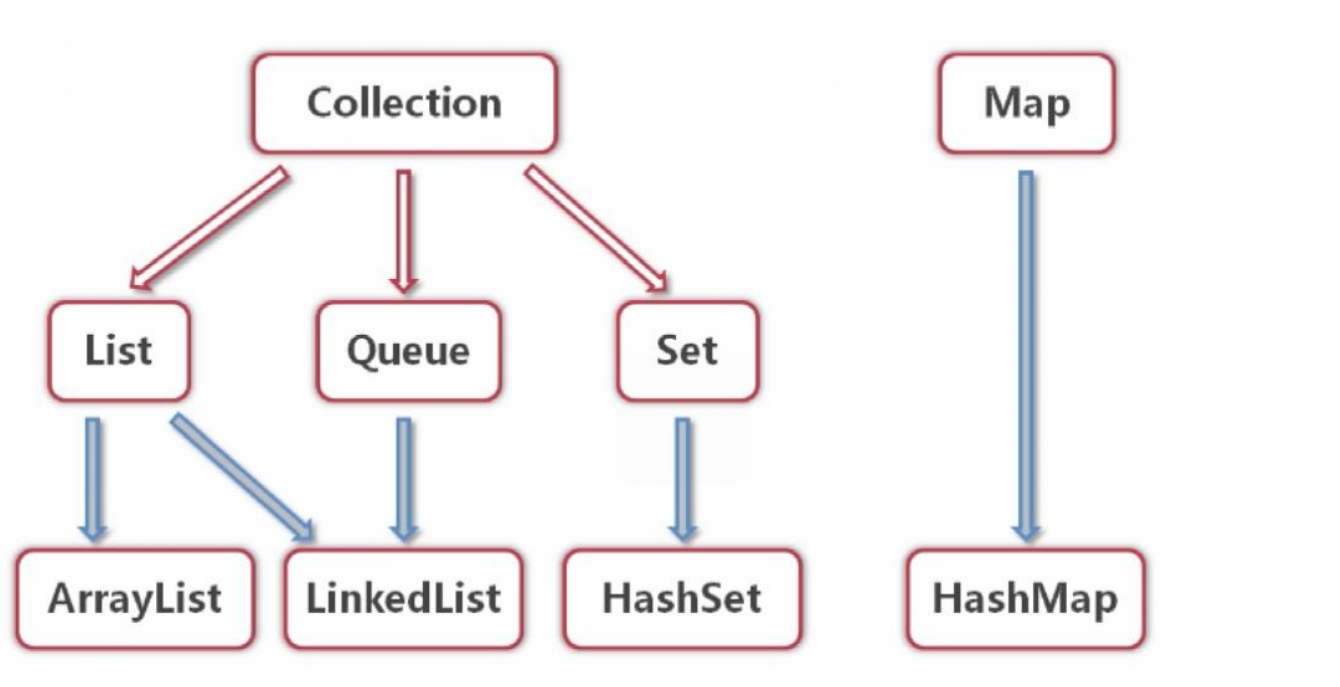
集合

- **疑问：为什么使用集合，而不用数组？**

应用场景

- 无法预测存储数据的数量
- 同时存储具有一对一关系的数据
- 需要进行数据的增删改查
- 数据重复问题

集合框架的体系结构



主要内容

- 如何创建String对象
- String对象的常用方法
- ==和equals方法的区别
- 字符串的不可变性

案例：公告管理

- 需求
  - 公告的添加和显示
  - 在指定位置处插入公告
  - 删除公告

集合排序

主要内容

- 集合中的基本数据类型排序
- 集合中的字符串排序
- Comparator接口
- Comparable接口

回顾

- **数组的排序**

```
//数组排序
int[] arr={15,31,23,46,65,12};
Arrays.sort(arr);
```

集合如何排序？

集合排序

- **使用Collections类的sort()方法**
- **sort(List<T> list)**

  -根据元素的自然顺序对指定列表按升序进行排序。

例题

- 例1：对存放在List中的整型数据进行排序。
- 例2：对存放在List中的字符串进行排序。

疑问：宠物猫如何排序？

解决办法：使用Comparable或Comparator接口。

Comparator接口

- 强行对某个对象进行整体排序的比较函数。
- 可以将 Comparator 传递给 sort 方法（如 Collections.sort 或 Arrays.sort）

Comparator接口

- int compare(T o1, T o2)　比较用来排序的两个参数。
  - 如果o1<o2，返回负整数
  - 如果o1==o2，返回0
  - 如果o1>o2，返回正整数

Comparator接口

- boolean equals(Object obj) 指示某个其他对象是否“等于”此Comparator
- 此方法可以被Object类中的equals方法覆盖，不必重写。

Comparator接口

- 例：对宠物猫分别按名字升序、年龄降序进行排列。

Comparable接口

- 此接口强行对实现它的每个类的对象进行整体排序。
- 这种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。
- 对于集合，通过调用Collections.sort方法进行排序。
- 对于数组，通过调用Arrays.sort方法进行排序。

Comparable接口

- int compareTo(T o)方法
- 该对象小于、等于或大于指定对象，则分别返回负整数、零或正整数。

- **例：对商品价格进行降序排列。**

Comparator和Comparable的区别

Comparator	Comparable