如何选择流中的元素：用谓词筛选，筛选出各不相同的元素，忽略流中的头几个元素，或将流截短至指定长度。

#### 1、用谓词筛选

Stream接口支持filter方法。

Stream<T> filter(Predicate<? super T> predicate);

该操作会接受一个谓词（一个返回boolean的函数）作为参数，并返回一个包括所有符合谓词的元素的流。

#### 2、筛选各异的元素

流还支持一个叫作distinct的方法，它会返回一个元素各异的流。

List<String> list = new ArrayList<>();

list.add("a");  
list.add("a");  
list.add("b");  
list.add("c");  
list.add("d");

List unique = list.stream()  
 .distinct()  
 .collect(Collectors.*toList*());



#### 3、截短流

流支持limit(n)方法，该方法会返回一个不超过给定长度的流。所需的长度作为参数传递给limit。如果流是有序的，则最多会返回前n个元素。

List<String> list = new ArrayList<>();

list.add("a");  
list.add("a");  
list.add("b");  
list.add("c");  
list.add("d");

List unique = list.stream()  
 .limit(2)  
 .collect(Collectors.*toList*());



注意：limit也可以用在无序流上，比如源是一个Set。这时，limit的结果不会以任何顺序排列。

#### 4、跳过元素

流还支持skip(n)方法，返回一个扔掉了前n个元素的流。如果流中元素不足n个，则返回一个空流。请注意，limit(n)和skip(n)是互补的！

List<String> list = new ArrayList<>();

list.add("a");  
list.add("a");  
list.add("b");  
list.add("c");  
list.add("d");

List unique = list.stream()  
 .skip(2)  
 .collect(Collectors.*toList*());

