Understand Java Streams Intermediate Operations dropWhile() & takeWhile()

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

- Understand the structure & functionality of stream aggregate operations
- Intermediate operations
  - map() & mapToInt()
  - filter() & flatMap()
  - dropWhile() & takeWhile()

These are both stateful, short-circuiting operations introduced in Java 9
Overview of the dropWhile() Intermediate Operation
Overview of the dropWhile() Intermediate Operation

- Overview of the dropWhile() intermediate operation

If stream is unordered, return a stream consisting of the remaining elements of this stream after dropping a subset of elements that match the given predicate.

See docs.oracle.com/javase/9/docs/api/java/util/stream/Stream.html#dropWhile
If stream is ordered, return a stream containing the remaining elements of this stream after dropping the longest prefix of elements matching the given predicate.

See docs.oracle.com/javase/9/docs/api/java/util/stream/Stream.html#dropWhile
Overview of the `dropWhile()` Intermediate Operation

- Overview of the `dropWhile()` intermediate operation

`dropWhile()` is a “stateful” operation that is costly on ordered parallel streams since threads must cooperate to find the longest contiguous sequence of matching elements in encounter order.

See blog.indrek.io/articles/whats-new-in-java-9-streams
Example of applying dropWhile() in the SimpleSearchStream program

The # of output stream elements may be less than the # of input stream elements.

However, the semantics of dropWhile() differ from the semantics of filter().
Overview of the `dropWhile()` Intermediate Operation

- Example of applying `dropWhile()` in the `SimpleSearchStream` program

```
List<SearchResults>
Stream<SearchResults>
Map<String, List<SearchResults>>
Stream<Entry<String, List<SearchResults>>>
Void

stream()
collect(groupingBy(...))
entrySet().stream()
dropWhile(e -> notEqual(e, word))
forEach(this::printResult)
```

`dropWhile()` also can’t change the type or values of elements it processes
Overview of the `dropWhile()` Intermediate Operation

- Example of applying `dropWhile()` in the `SimpleSearchStream` program

```java
listOfResults.stream()
  .collect(groupingBy
    (SearchResults::getWord,
     LinkedHashMap::new,
     toList()))
  .entrySet().stream()
  .dropWhile(e -> notEqual(e, word))
  .forEach(e -> printResult
    (e.getKey(),
     e.getValue()));
```
Example of applying `dropWhile()` in the SimpleSearchStream program

```java
listOfResults
    .stream()
    .collect
    (groupingBy
        (SearchResults::getWord,
         LinkedHashMap::new,
         toList()))
    .entrySet()
    .stream()
    .dropWhile(e -> notEqual(e, word))
    .forEach(e -> printResult
        (e.getKey(),
         e.getValue()));
```
Overview of the dropWhile() Intermediate Operation

- Example of applying dropWhile() in the SimpleSearchStream program

```java
listOfResults
  .stream()
  .collect((groupingBy
    (SearchResults::getWord,
      LinkedHashMap::new,
      toList()))
  .entrySet()
  .stream()
  .dropWhile(e -> notEqual(e, word))
  .forEach(e -> printResult
    (e.getKey(),
     e.getValue()));
```

Collect stream into a map with words as key

```
stream()
    collect(groupingBy(...))
    entrySet().stream()
    dropWhile(e -> notEqual(e, word))
    forEach(this::printResult)
```
Overview of the dropWhile() Intermediate Operation

- Example of applying dropWhile() in the SimpleSearchStream program

```java
listOfResults.stream()
    .collect((groupingBy
        (SearchResults::getWord,
         LinkedHashMap::new,
         toList()))
    .entrySet()
    .stream()
    .dropWhile(e -> notEqual(e, word))
    .forEach(e -> printResult
        (e.getKey(),
         e.getValue()));
```
Overview of the dropWhile() Intermediate Operation

- Example of applying dropWhile() in the SimpleSearchStream program

```java
listOfResults.stream() .collect((groupingBy (SearchResults::getWord, LinkedHashMap::new, toList()))) .entrySet().stream() .dropWhile(e -> notEqual(e, word)) .forEach(e -> printResult(e.getKey(), e.getValue()));
```

*notEqual() is defined as return !e.getKey().equals(word)*
Overview of the dropWhile() Intermediate Operation

- Example of applying dropWhile() in the SimpleSearchStream program

```java
listOfResults
  .stream()
  .collect
    (groupingBy
       (SearchResults::getWord,
        LinkedHashMap::new,
        toList()))
  .entrySet()
  .stream()
  .dropWhile(e -> notEqual(e, word))
  .forEach(e -> printResult
            (e.getKey(),
             e.getValue()));
```

Print results starting at the match
Overview of the takeWhile() Intermediate Operation
Overview of the `takeWhile()` Intermediate Operation

- Overview of the `takeWhile()` intermediate operation

If this stream is unordered then return a stream consisting of a subset of elements taken from this stream that match the given predicate.

See [docs.oracle.com/javase/9/docs/api/java/util/stream/Stream.html#takeWhile](https://docs.oracle.com/javase/9/docs/api/java/util/stream/Stream.html#takeWhile)
If this stream is ordered then return a stream consisting of the longest prefix of elements taken from this stream that match the given predicate.

See docs.oracle.com/javase/9/docs/api/java/util/stream/Stream.html#takeWhile
Overview of the `takeWhile()` Intermediate Operation

- Overview of the `takeWhile()` intermediate operation

`takeWhile()` is a "stateful" operation that is costly on ordered parallel streams since threads must cooperate to find the longest contiguous sequence of matching elements in encounter order.

See blog.indrek.io/articles/whats-new-in-java-9-streams
Example of applying `takeWhile()` in the SimpleSearchStream program

The # of output stream elements may be less than the # of input stream elements.

However, the semantics of `takeWhile()` differ from the semantics of `filter()`.

```
stream()
collect(groupingBy(…))
entrySet().stream()
takeWhile(e -> notEqual(e, word))
forEach(this::printResult)
```
Overview of the `takeWhile()` Intermediate Operation

- Example of applying `takeWhile()` in the `SimpleSearchStream` program

```
List <SearchResults>

Stream <SearchResults>

Map<String, List <SearchResults>

Stream<Entry<String, List <SearchResults> >>

Stream<Entry<String, List <SearchResults> >>

Void
```

takeWhile() also can’t change the type or values of elements it processes
Overview of the takeWhile() Intermediate Operation

- Example of applying takeWhile() in the SimpleSearchStream program

```java
listOfResults
  .stream()
  .collect
    (groupingBy
      (SearchResults::getWord,
       LinkedHashMap::new,
       toList())
    .entrySet()
  .stream()
  .takeWhile(e -> notEqual(e, word))
  .forEach(e -> printResult
    (e.getKey(),
     e.getValue()));
```
Overview of the `takeWhile()` Intermediate Operation

- Example of applying `takeWhile()` in the SimpleSearchStream program

```java
listOfResults
  .stream()
  .collect
    (groupingBy
      (SearchResults::getWord,
       LinkedHashMap::new,
      .toList()))
  .entrySet()
  .stream()
  .takeWhile(e -> notEqual(e, word))
  .forEach(e -> printResult
    (e.getKey(),
     e.getValue()));
```

Convert list of search results into a stream
Overview of the `takeWhile()` Intermediate Operation

- Example of applying `takeWhile()` in the SimpleSearchStream program

```
listOfResults
  .stream()
  .collect
    (groupingBy
      (SearchResults::getWord, 
       LinkedHashMap::new, 
       toList()))
  .entrySet()
  .stream()
  .takeWhile(e -> notEqual(e, word))
  .forEach(e -> printResult
    (e.getKey(),
     e.getValue()));
```

Collect stream into a map with words as key
Overview of the `takeWhile()` Intermediate Operation

- Example of applying `takeWhile()` in the SimpleSearchStream program

```java
takeWhile(e -> notEqual(e, word)).forEach(e -> printResult(e.getKey(), e.getValue()));
```
Overview of the `takeWhile()` Intermediate Operation

- Example of applying `takeWhile()` in the SimpleSearchStream program

```java
listOfResults
    .stream()
    .collect((groupingBy
        (SearchResults::getWord,
         LinkedHashMap::new,
         toList()))
    .entrySet()
    .stream()
    .takeWhile(e -> notEqual(e, word))
    .forEach(e -> printResult(e.getKey(), e.getValue()));
```

`notEqual()` is defined as `return !e.getKey().equals(word)`
Overview of the `takeWhile()` Intermediate Operation

- Example of applying `takeWhile()` in the `SimpleSearchStream` program

```java
listOfResults
  .stream()
  .collect
    (groupingBy
      (SearchResults::getWord,
        LinkedHashMap::new,
        toList()))
  .entrySet()
  .stream()
  .takeWhile(e -> notEqual(e, word))
  .forEach(e -> printResult
    (e.getKey(),
    e.getValue()));
```

Print results starting at the beginning & continuing up to (but not including) the match
Intermediate Operations

dropWhile() & takeWhile()