Java Streams: Implementing WordSearcher.printResults()

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

- Visualize aggregate operations in SimpleSearchStream's WordSearcher .printResults() method
- Understand the implementation of aggregate operations in SimpleSearch Stream's WordSearcher.printResults() method

```
public void printResults(List<SearchResults> listOfResults) {
  listOfResults
    .stream()

    .collect(groupingBy(SearchResults::getWord,
                        LinkedHashMap::new,
                        toDownstreamCollector()))

    .forEach(this::printResult);
}
```

This lesson shows the collect(groupingBy()) & mapToInt() aggregate operations

Implementing the Word Searcher.printResults() Method

Implementing the WordSearcher.printResults() Method

- This method prints the results of the word search

```
public void printResults(List<SearchResults> listOfResults) {
  listOfResults
    .stream()



    .collect(groupingBy(SearchResults::getWord,
                        LinkedHashMap::new,
                        toDownstreamCollector()))


    .forEach(this::printResult);
}
```

Implementing the WordSearcher.printResults() Method

- This method prints the results of the word search

```
public void printResults(List<SearchResults> listOfResults) {
  listOfResults
    .stream()



    .collect(groupingBy(SearchResults::getWord,
                        LinkedHashMap::new,
                        toDownstreamCollector()))


    .forEach(this::printResult);
}
```

Convert the list param into a stream.

Implementing the WordSearcher.printResults() Method

- This method prints the results of the word search

```
public void printResults(List<SearchResults> listOfResults) {
  listOfResults
    .stream()



    .collect(groupingBy(SearchResults::getWord,
                        LinkedHashMap::new,
                        toDownstreamCollector()))


    .forEach(this::printResult);
}
```

Collect SearchResults into a Map, with word as the key & the list of indices as the value.

Implementing the WordSearcher.printResults() Method

- This method prints the results of the word search

```
public void printResults(List<SearchResults> listOfResults) {
  listOfResults
    .stream()



    .collect(groupingBy(SearchResults::getWord,
                        LinkedHashMap::new,
                        toDownstreamCollector()))


    .forEach(this::printResult);
}
```

LinkedHashMap preserves the insertion order wrt iteration.

See docs.oracle.com/javase/8/docs/api/java/util/LinkedHashMap.html

Implementing the WordSearcher.printResults() Method

- This method prints the results of the word search

```
public void printResults(List<SearchResults> listOfResults) {
  listOfResults
    .stream()



    .collect(groupingBy(SearchResults::getWord,
                        LinkedHashMap::new,
                        toDownstreamCollector()))



    .forEach(this::printResult);
}
```

This factory method creates a downstream collector that merges results lists together.

See upcoming lesson on "*Java Streams: Applying Non-Concurrent Collectors*"

Implementing the WordSearcher.printResults() Method

- This method prints the results of the word search

```
public void printResults(List<SearchResults> listOfResults) {
  listOfResults
    .stream()




    .collect(groupingBy(SearchResults::getWord,
                        LinkedHashMap::new,
                        toDownstreamCollector()))



    .forEach(this::printResult);
}
```

Print out the matching results in the stream.

This is the Map forEach() method *not* the Stream forEach() method!

Implementing the WordSearcher.printResults() Method

- Print a word and its list of indices to the output

```
private void printResult(String word,
                         List<SearchResults.Result> results) {
  System.out.print("Word \""
                   + word
                   + "\" appeared at indices ");



  SearchResults.printResults(results);

  System.out.println(" with max index of "
               + computeMax(results));
}
```

Implementing the WordSearcher.printResults() Method

- Print a word and its list of indices to the output

```
private void printResult(String word,
                         List<SearchResults.Result> results) {
  System.out.print("Word \""
                   + word
                   + "\" appeared at indices ");

  SearchResults.printResults(results);

  System.out.println(" with max index of "
              + computeMax(results));
}
```

Print the word followed by the list of search results.

Implementing the WordSearcher.printResults() Method

- Print a word and its list of indices to the output

```
private void printResult(String word,
                         List<SearchResults.Result> results) {
  System.out.print("Word \""
                   + word
                   + "\" appeared at indices ");




  SearchResults.printResults(results);

  System.out.println(" with max index of "
                + computeMax(results));
}
```

Compute & print the max index.

Implementing the WordSearcher.printResults() Method

- Compute the max index in the list of search results

```
private int computeMax(List<SearchResults.Result> results) {
  return results
    .stream()


    .mapToInt(SearchResults.Result::getIndex)


    .max()


    .orElse(0);
}
```

Ths implementation works properly even if the results are not sorted!

Implementing the WordSearcher.printResults() Method

- Compute the max index in the list of search results

```
private int computeMax(List<SearchResults.Result> results) {
  return results
    .stream()


    .mapToInt(SearchResults.Result::getIndex)


    .max()


    .orElse(0);
}
```

Convert the list results into a stream of results

Implementing the WordSearcher.printResults() Method

- Compute the max index in the list of search results

```
private int computeMax(List<SearchResults.Result> results) {
  return results
    .stream()


    .mapToInt(SearchResults.Result::getIndex)


    .max()


    .orElse(0);
}
```

Map the stream of Result objects into a stream of int primitives.

See docs.oracle.com/javase/8/docs/api/java/util/stream/Stream.html#mapToInt

Implementing the WordSearcher.printResults() Method

- Compute the max index in the list of search results

```
private int computeMax(List<SearchResults.Result> results) {
  return results
    .stream()


    .mapToInt(SearchResults.Result::getIndex)


    .max()


    .orElse(0);
}
```

Returns an OptionalInt describing the maximum element of this stream or an empty optional if this stream is empty

OPTIONAL

See docs.oracle.com/javase/8/docs/api/java/util/stream/IntStream.html#max

Implementing the WordSearcher.printResults() Method

- Compute the max index in the list of search results

```
private int computeMax(List<SearchResults.Result> results) {
  return results
    .stream()

    .mapToInt(SearchResults.Result::getIndex)

    .max()

    .orElse(0);
}
```

Return the value (as an int) if present, otherwise return 0.

OPTIONAL

See docs.oracle.com/javase/8/docs/api/java/util/OptionalInt.html#orElse

End of Java Streams: Implementing Word Searcher.printResults()