

Overview of the Simple SearchStream Program

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

**Institute for Software
Integrated Systems**

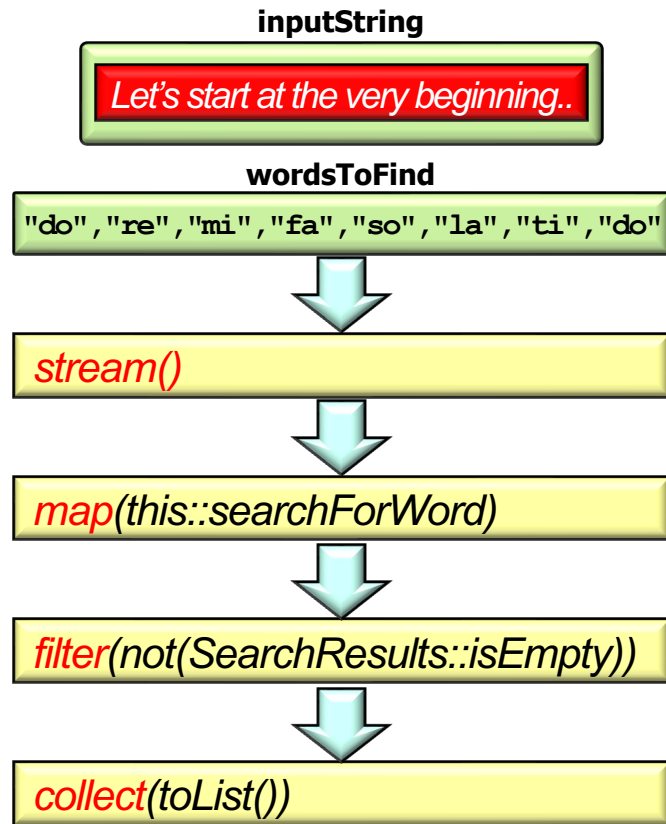
**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

- Understand the structure & functionality of the SimpleSearchStream program

We use this program to showcase key Java sequential streams capabilities.

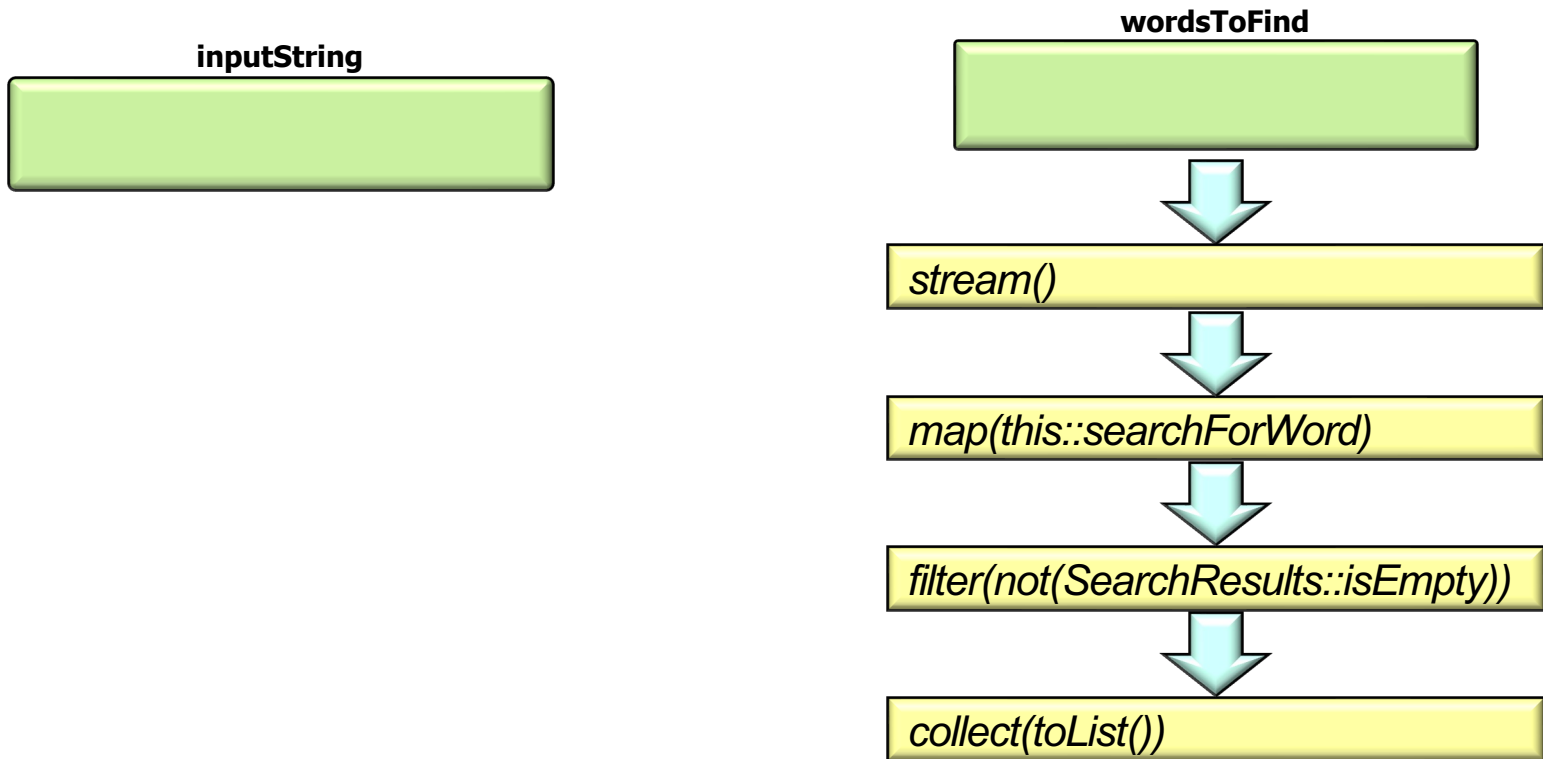


See github.com/douglasraigschmidt/LiveLessons/tree/master/SimpleSearchStream

Visualizing the Simple SearchStream program

Visualizing the SimpleSearchStream Program

- This program finds words in an input string



See github.com/douglasraigschmidt/LiveLessons/tree/master/SimpleSearchStream

Visualizing the SimpleSearchStream Program

- This program finds words in an input string

inputString

Let's start at the very beginning...



wordsToFind

"do", "re", "mi", "fa",
"so", "la", "ti", "do"

`stream()`

`map(this::searchForWord)`

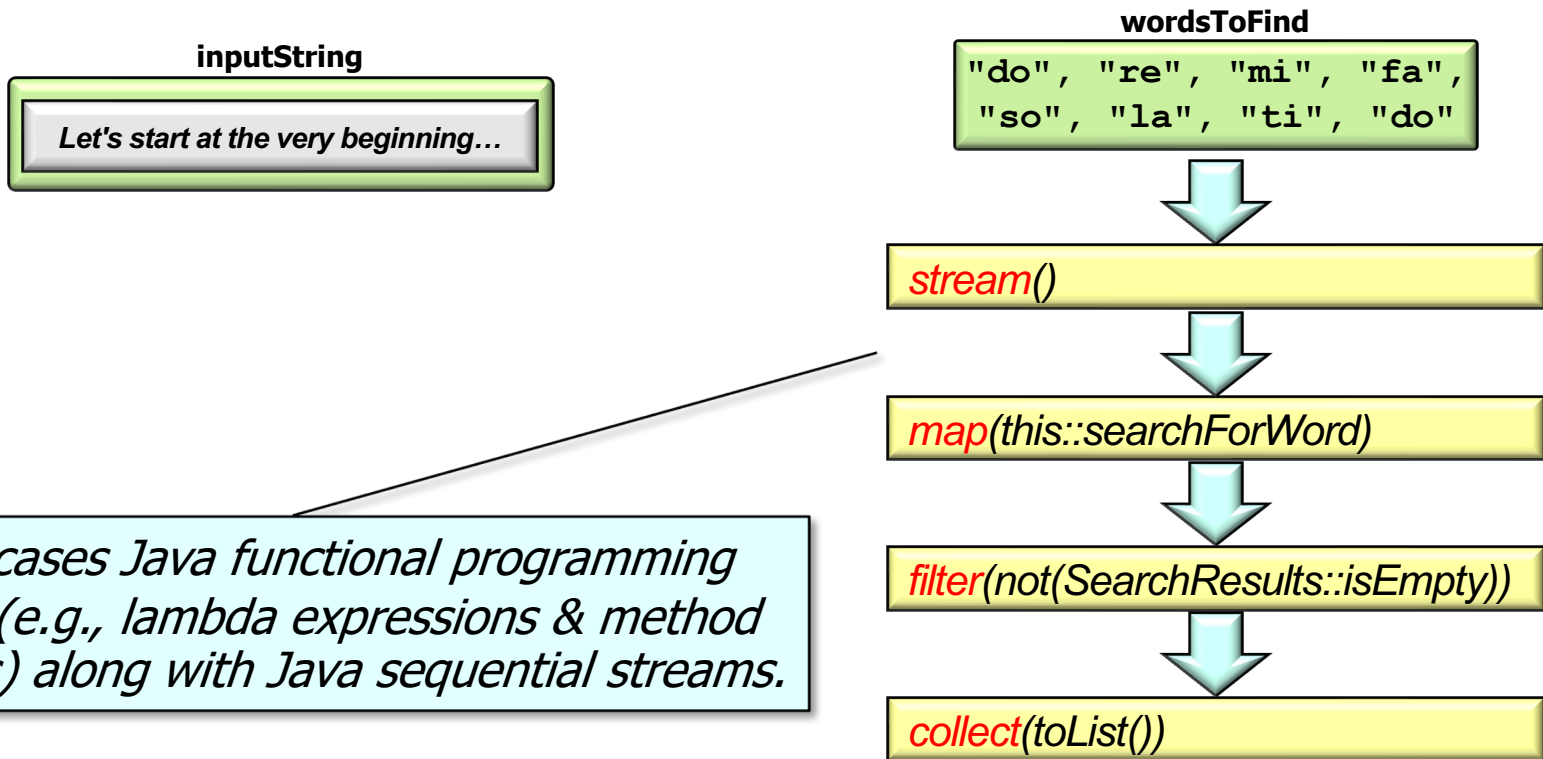
`filter(not(SearchResults::isEmpty))`

`collect(toList())`

See en.wikipedia.org/wiki/Do-Re-Mi

Visualizing the SimpleSearchStream Program

- This program finds words in an input string



See [SimpleSearchStream/src/main/java/search/WordSearcher.java](https://github.com/akkaio/simple-search-stream/blob/master/src/main/java/search/WordSearcher.java)

Visualizing the SimpleSearchStream Program

- This program finds words in an input string

inputString

Let's start at the very beginning...

Starting SimpleSearchStream

Word "Re" matched at index [131|141|151|202|212|222|
979|1025|1219|1259|
1278|1300|1351|1370|1835|
1875|1899|1939|2266|2295|
Word "Ti" matched at index [237|994|1272|1294|1364|1850|
1860|1912|1915|1952|1955|
2299|
Word "La" matched at index [234|417|658|886|991|1207|
1247|1269|1291|1339|1361|
1742|1847|1863|1909|1949|
2161|2254|2276|2283]...

Ending SimpleSearchStream

wordsToFind

"do", "re", "mi", "fa",
"so", "la", "ti", "do"

`stream()`

`map(this::searchForWord)`

`filter(not(SearchResults::isEmpty))`

`collect(toList())`

The program produces nicely formatted output.

Visualizing the SimpleSearchStream Program

- It also prints a slice of search results starting at a particular word, e.g., "La"

inputString

Let's start at the very beginning...

Starting SimpleSearchStream

...

Word "La" appeared at indices

[234|417|658|886|991|1207|1247|1269|1291|1339|
1361|1742|1847|1863|1909|1949|2161|2254|2276|2283]

Word "Ti" appeared at indices

[237|994|1272|1294|1364|1850|1860|1912|1915|
1952|1955|2299]

... Ending SimpleSearchStream



`stream()`

`collect(groupingBy(...))`

`entrySet().stream()`

`dropWhile(e -> notEqual(e, word))`

`forEach(this::printResult)`

Print out results of each map entry (key = word & value = list of search results).

Entry Point Into the Simple SearchStream Program

Entry Point Into the SimpleSearchStream Program

- It searches sequentially for words in a string containing the contents of a file

```
static public void main(String[] args) { ...  
    String input = TestDataFactory  
        .getInput(sINPUT_FILE, "@").get(0);  
  
    List<String> wordsToFind = TestDataFactory  
        .getWordList(sWORD_LIST_FILE);  
  
    WordSearcher searcher =  
        new WordSearcher(input);  
  
    List<SearchResults> results =  
        searcher.findWords(wordsToFind);  
  
    searcher.printResults(results); ...
```

See SimpleSearchStream/src/main/java/Main.java

Entry Point Into the SimpleSearchStream Program

- It searches sequentially for words in a string containing the contents of a file

```
static public void main(String[] args) { ...
```

```
    String input = TestDataFactory  
        .getInput(sINPUT_FILE, "@").get(0);
```

```
    List<String> wordsToFind = TestDataFactory  
        .getWordList(sWORD_LIST_FILE);
```

*Create an input string containing
the lyrics to the do-re-mi song.*

```
    WordSearcher searcher =  
        new WordSearcher(input);
```

```
    List<SearchResults> results =  
        searcher.findWords(wordsToFind);
```

```
    searcher.printResults(results); ...
```

See SimpleSearchStream/src/main/java/utils/TestDataFactory.java

Entry Point Into the SimpleSearchStream Program

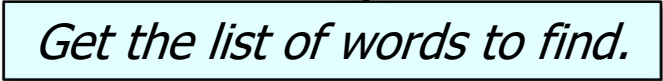
- It searches sequentially for words in a string containing the contents of a file

```
static public void main(String[] args) { ...
```

```
    String input = TestDataFactory  
        .getInput(sINPUT_FILE, "@").get(0);
```

```
    List<String> wordsToFind = TestDataFactory  
        .getWordList(sWORD_LIST_FILE);
```

```
    WordSearcher searcher =  
        new WordSearcher(input);
```



Get the list of words to find.

```
    List<SearchResults> results =  
        searcher.findWords(wordsToFind);
```

```
    searcher.printResults(results); ...
```

See [SimpleSearchStream/src/main/java/Utils/TestDataFactory.java](https://github.com/AlvinChen/SimpleSearchStream/blob/master/src/main/java/Utils/TestDataFactory.java)

Entry Point Into the SimpleSearchStream Program

- It searches sequentially for words in a string containing the contents of a file

```
static public void main(String[] args) { ...
```

```
    String input = TestDataFactory  
        .getInput(sINPUT_FILE, "@").get(0);
```

```
    List<String> wordsToFind = TestDataFactory  
        .getWordList(sWORD_LIST_FILE);
```

```
    WordSearcher searcher =  
        new WordSearcher(input);
```

*Create an object used to search
for words in the input string.*

```
    List<SearchResults> results =  
        searcher.findWords(wordsToFind);
```

```
    searcher.printResults(results); ...
```

See SimpleSearchStream/src/main/java/search/WordSearcher.java

Entry Point Into the SimpleSearchStream Program

- It searches sequentially for words in a string containing the contents of a file

```
static public void main(String[] args) { ...  
    String input = TestDataFactory  
        .getInput(sINPUT_FILE, "@").get(0);
```

```
    List<String> wordsToFind = TestDataFactory  
        .getWordList(sWORD_LIST_FILE);
```

```
    WordSearcher searcher =  
        new WordSearcher(input);
```

Find all matching words.

```
    List<SearchResults> results =  
        searcher.findWords(wordsToFind);
```

```
    searcher.printResults(results); ...
```

Entry Point Into the SimpleSearchStream Program

- It searches sequentially for words in a string containing the contents of a file

```
static public void main(String[] args) { ...  
    String input = TestDataFactory  
        .getInput(sINPUT_FILE, "@").get(0);
```

```
    List<String> wordsToFind = TestDataFactory  
        .getWordList(sWORD_LIST_FILE);
```

```
    WordSearcher searcher =  
        new WordSearcher(input);
```

```
    List<SearchResults> results =  
        searcher.findWords(wordsToFind);
```

Print all matching words.

```
    searcher.printResults(results); ...
```

End of Overview of the SimpleSearchStream Program