Overview of the Simple SearchStream Program

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

Professor of Computer Science

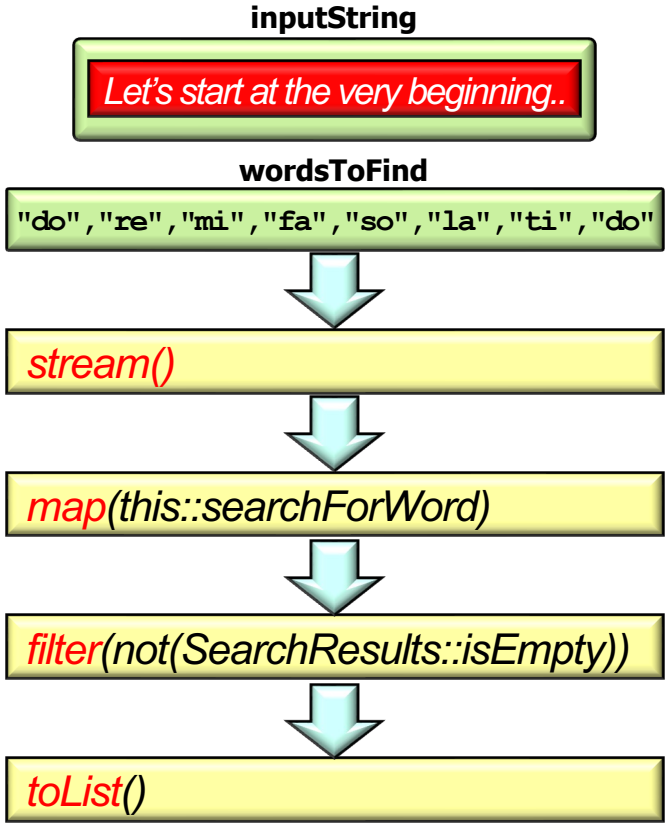
Institute for Software Integrated Systems

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.

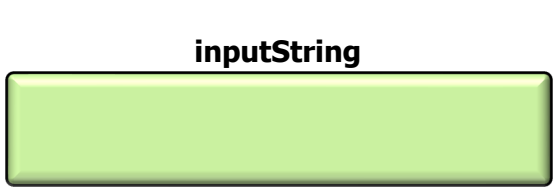


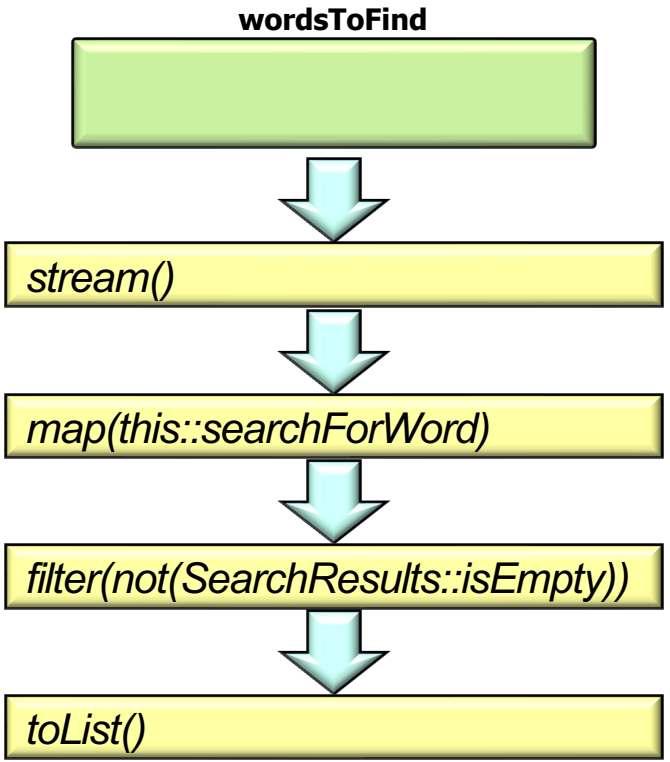
See github.com/douglascraigschmidt/LiveLessons/tree/master/SimpleSearchStream

Visualizing the Simple SearchStream program

Visualizing the SimpleSearchStream Program

- This program finds words in an input string

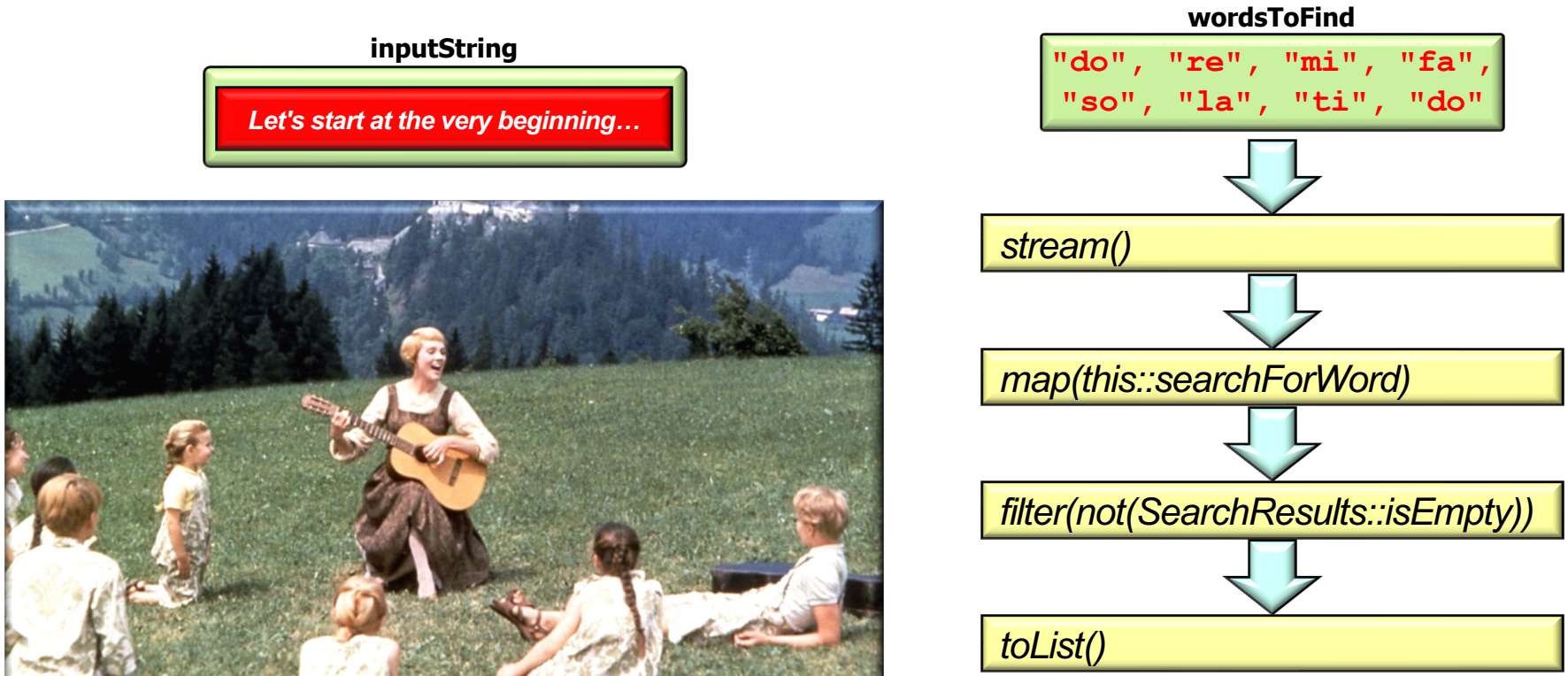




See github.com/douglascraigschmidt/LiveLessons/tree/master/SimpleSearchStream

Visualizing the SimpleSearchStream Program

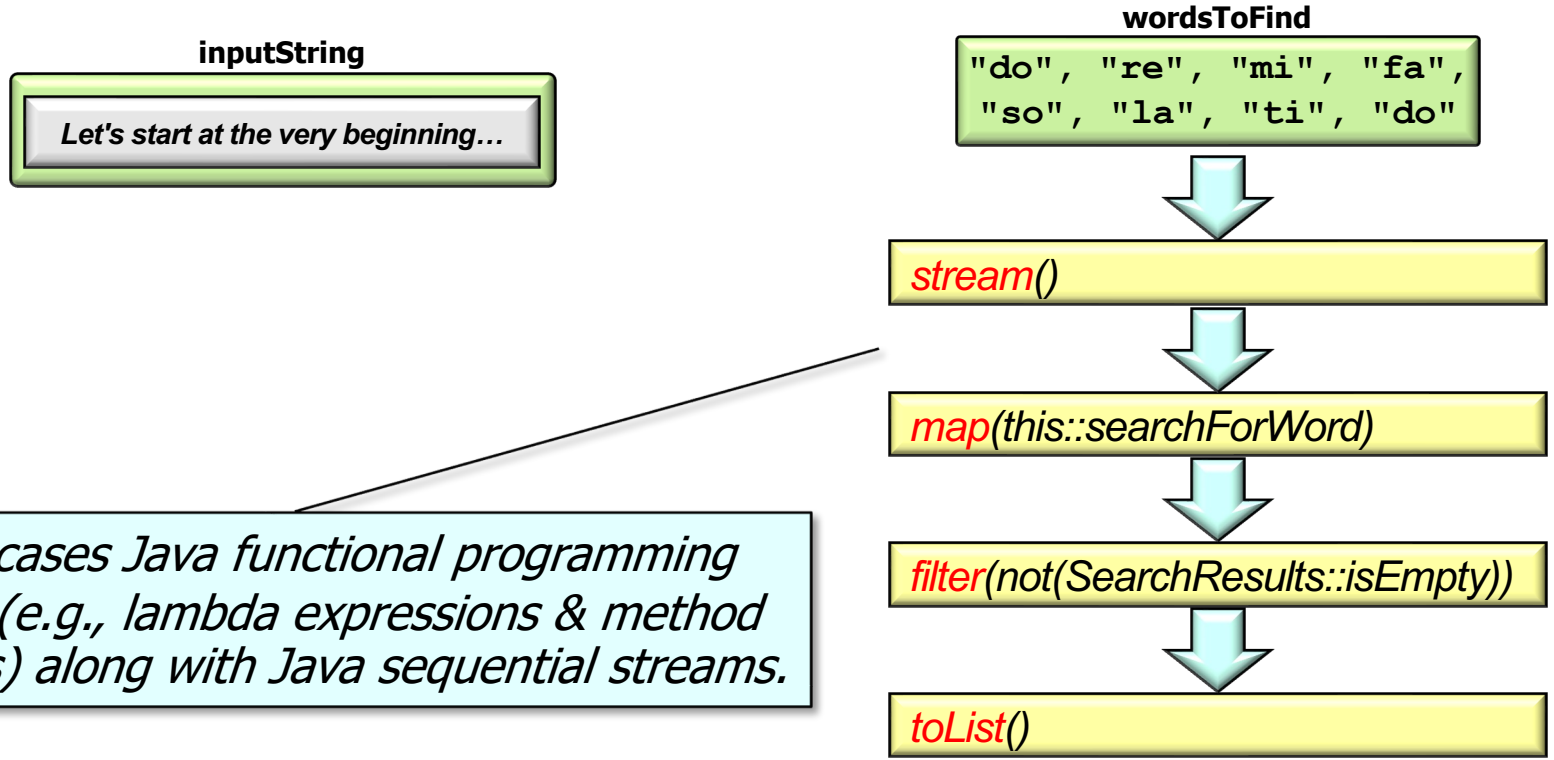
- This program finds words in an input string



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

Visualizing the SimpleSearchStream Program

- This program finds words in an input string

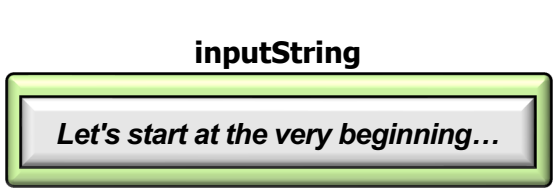


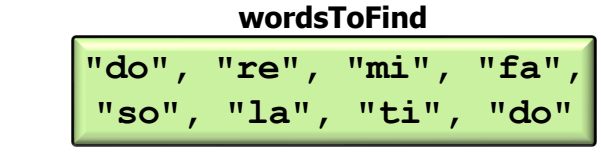
It showcases Java functional programming features (e.g., lambda expressions & method references) along with Java sequential streams.

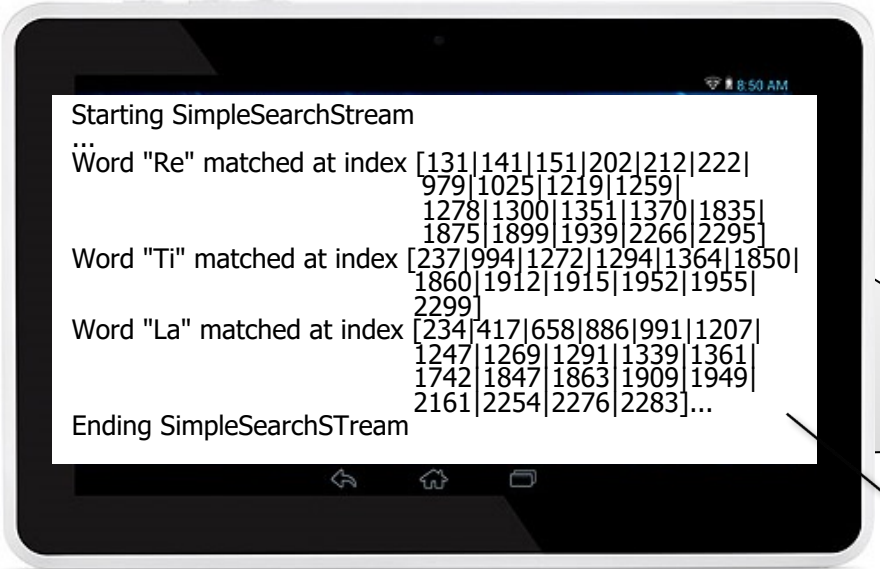
See SimpleSearchStream/src/main/java/search/WordSearcher.java

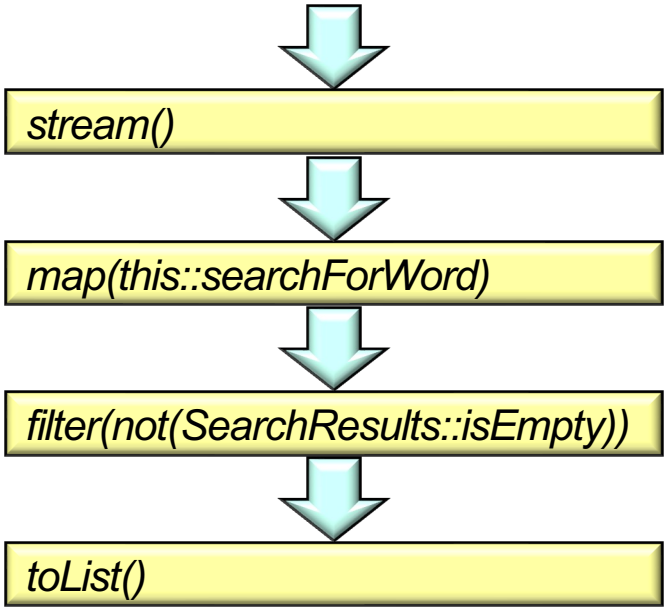
Visualizing the SimpleSearchStream Program

- This program finds words in an input string





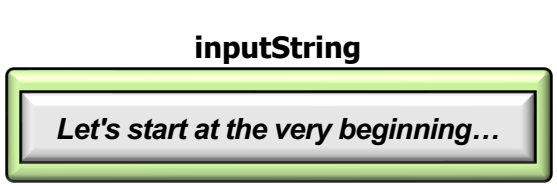


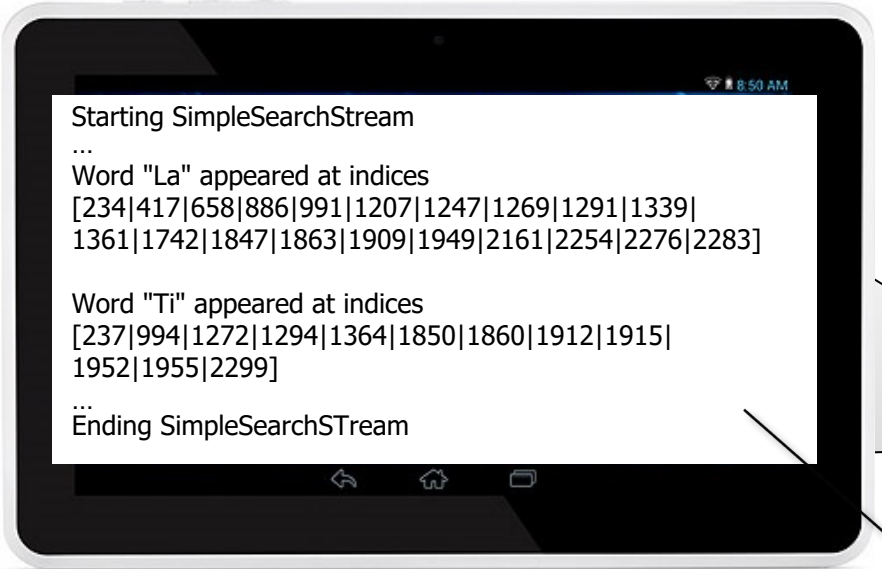


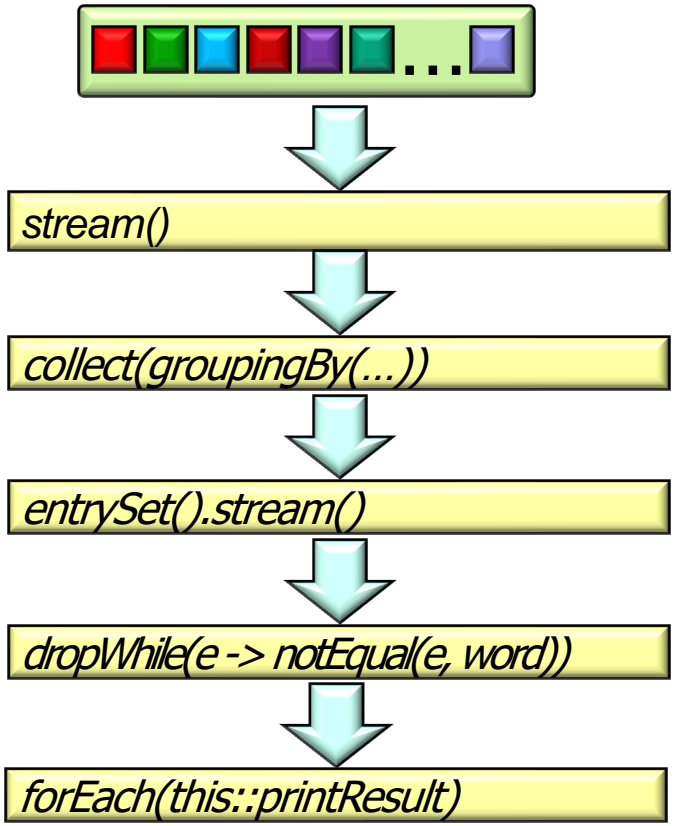
The program produces nicely formatted output.

Visualizing the SimpleSearchStream Program

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







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); ...
```

This driver code doesn't use streams

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);

  WordSearcher searcher =
    new WordSearcher(input);

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

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

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

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);

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

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

Get the list of words to find.

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);

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

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

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

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

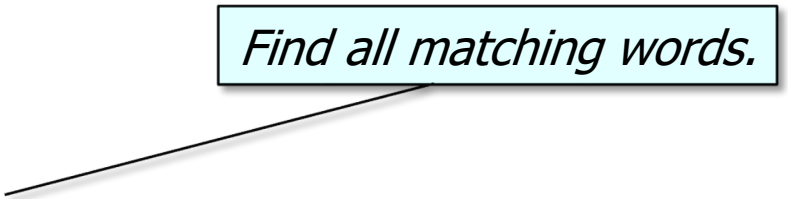
```
static public void main(String[] args) { ...
  String input = TestDataFactory
    .getInput(sINPUT_FILE, "@").get(0);

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

  WordSearcher searcher =
    new WordSearcher(input);

  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

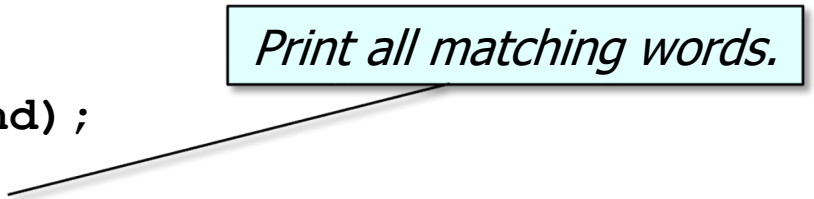
```
static public void main(String[] args) { ...
  String input = TestDataFactory
    .getInput(sINPUT_FILE, "@").get(0);

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

  WordSearcher searcher =
    new WordSearcher(input);

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

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



End of Overview of the SimpleSearchStream Program