Java Sequential SearchStreamGang

Example: Visualizing Hook Methods

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

- Know how to apply sequential streams to the SearchStreamGang program
- Understand the SearchStreamGang process Stream() & processInput() hook methods

Starting SearchStreamGangTest
PARALLEL_SPLITTERATOR executed in 409 msecs
COMPLETABLE_FUTURES_INPUTS executed in 426 msecs
COMPLETABLE_FUTURES_PHASES executed in 427 msecs
PARALLEL_STREAMS executed in 437 msecs
PARALLEL_STREAM_PHASES executed in 440 msecs
RXJAVA_PHASES executed in 485 msecs
PARALLEL_STREAM_INPUTS executed in 802 msecs
RXJAVA_INPUTS executed in 866 msecs
SEQUENTIAL_LOOPS executed in 1638 msecs
SEQUENTIAL_STREAM executed in 1958 msecs
Ending SearchStreamGangTest

Search Phrases

Stream(phrases -> searchForPhrase(…))

filter(not(SearchResults::isEmpty))

collect(toList())

45,000+ phrases
Search Phrases
Visualizing the processStream() Hook Method
Visualizing the processStream() Hook Method

- processStream() searches a list of input strings

```
List<String>
```

- Input a list of input strings

```
stream()
```
Visualizing the `processStream()` Hook Method

- `processStream()` searches a list of input strings

```
List<String>
```

Convert collection to a (sequential) stream
Visualizing the `processStream()` Hook Method

- `processStream()` searches a list of input strings

```
List <String> ...
```

```
Stream <String> ...
```

Output a stream of input strings

Input Strings to Search

`stream()`
Visualizing the `processStream()` Hook Method

- `processStream()` searches a list of input strings.

```
List <String>…
stream()…
Stream <String>
map(this::processInput)
```

Input Strings to Search
Visualizing the `processStream()` Hook Method

- `processStream()` searches a list of input strings

List `<String>`

Stream `<String>`

Call `processInput()` to search for phrases in each input string
Visualizing the `processStream()` Hook Method

- `processStream()` searches a list of input strings

Output a stream of lists of search results

List `<String>`

Stream `<String>`

Stream `<List<SearchResults>>`

Input Strings to Search

`stream()`

`map(this::processInput)`

List of search results
Visualizing the `processStream()` Hook Method

- `processStream()` searches a list of input strings

```
Output a stream of lists of search results
```

```
List <String>
```

```
Stream <String>
```

```
Stream<List<SearchResults>>
```

```
map() also transforms its input stream type into a different output stream type
```
processStream() searches a list of input strings

Input Strings to Search

List<String>

Stream<String>

Stream<List<SearchResults>>

Input a stream of lists of search results
Visualizing the `processStream()` Hook Method

- `processStream()` searches a list of input strings

List `<String>`

Stream `<String>`

Stream `<List<SearchResults>>`  

Trigger intermediate operation processing
**Visualizing the processStream() Hook Method**

- `processStream()` searches a list of input strings

```
List<String>...
```

```
Stream<String>...
```

```
Stream<List<SearchResults>>...
```

```
List<List<SearchResults>>...
```

**Return a list of lists of search results based on “encounter order”**
Visualizing the processInput() Hook Method
Visualizing the processInput() Hook Method

- processInput() finds phrases in an input string

Input a list of phrases to find

List <String>

... stream()

Search Phrases
Visualizing the `processInput()` Hook Method

- `processInput()` finds phrases in an input string

List `<String>`

`stream()` Convert collection to a (sequential) stream
• processInput() finds phrases in an input string

Output a stream of phrases to find

List <String>

Stream <String>

Search Phrases

stream()
• `processInput()` finds phrases in an input string

```
Input a stream of phrases to find
```

```
List <String> ...
```

```
Stream <String> ...
```

```
stream()
```

```
map(phrase -> searchForPhrase(...))
```

Visualizing the `processInput()` Hook Method

45,000+ phrases

List <String>

Stream <String>

Search Phrases
Visualizing the `processInput()` Hook Method

- `processInput()` finds phrases in an input string

```
List <String>

Stream <String>
```

Search for the phrase in each input string

```
stream()

map(phrase -> searchForPhrase(...))
```
Visualizing the `processInput()` Hook Method

- `processInput()` finds phrases in an input string

List `<String>`

Stream `<String>`

Stream `<SearchResults>`

Output a stream of search results
Visualizing the `processInput()` Hook Method

- `processInput()` finds phrases in an input string

```
List
<String>

Stream
<String>

Stream
<SearchResults>
```

Input a stream of search results

Search Phrases

```
stream()

map(phrase -> searchForPhrase(...))

filter(not(SearchResults::isEmpty))
```
Visualizing the `processInput()` Hook Method

- `processInput()` finds phrases in an input string

```
List <String>

Stream <String>

Stream <SearchResults>
```

- Remove empty search results from the stream

```
stream()

map(phrase -> searchForPhrase(…))

filter(not(SearchResults::isEmpty))
```
Visualizing the processInput() Hook Method

- **processInput()** finds phrases in an input string

Output a stream of non-empty search results

List `<String>`

Stream `<String>`

Stream `<SearchResults>`

Stream `<SearchResults>`

**Search Phrases**

- `stream()`
- `map(phrase -> searchForPhrase(...))`
- `filter(not(SearchResults::isEmpty))`
Visualizing the processInput() Hook Method

- processInput() finds phrases in an input string

**Input a stream of non-empty search results**

```
List <String>
```

```
Stream <String>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
```

```
Stream <SearchResults>
Visualizing the `processInput()` Hook Method

- `processInput()` finds phrases in an input string

```
List <String>
```

```
Stream <String>
```

```
Stream <SearchResults>
```

Trigger intermediate operation processing

```
Search Phrases
```

```
stream()
```

```
map(phrase -> searchForPhrase(...))
```

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

```
collect(toList())
```

45,000+ phrases

Search Phrases
Visualizing the `processInput()` Hook Method

- `processInput()` finds phrases in an input string

```
List <String> …
Stream <String> …
Stream <SearchResults> …
List <SearchResults>
```

Return a (possibly empty) list of search results in encounter order
processStream() searches a list of input strings

Output a stream of lists of search results

List <String>

Stream <String>

Some lists of search results may be empty if no phrases match an input string
Visualizing the `processInput()` Hook Method

- `processStream()` searches a list of input strings

```
List<String>
```

```
Stream<String>
```

```
Stream<List<SearchResults>>
```

```
List<List<SearchResults>>
```

See "Java 8 Sequential SearchStreamGang Example (Part 2)"

Input Strings to Search

```
stream()
```

```
map(this::processInput)
```

```
collect(toList())
```

The `printPhrases()` method handles empty "list of search results" later
Visualizing the `processInput()` Hook Method

- We focus on sequential streams with one thread

```
List <String>  ➔
Stream <String>
Stream <SearchResults>
List <SearchResults>
```

```
Map(phrase -> searchForPhrase(...))
filter(not(SearchResults::isEmpty))
collect(toList())
```

45,000+ phrases Search Phrases

We focus on sequential streams with one thread.
Visualizing the `processInput()` Hook Method

- We focus on sequential streams with one thread
- We’ll cover parallel streams later

```
List <String>

Stream <String>

Stream <SearchResults>

List <SearchResults>
```

See “Overview of Java 8 Parallel Streams”
Visualizing the processInput() Hook Method

- We focus on sequential streams with one thread
- We’ll cover parallel streams later

Minuscule changes are needed to transition from sequential to parallel streams!
End of Java Sequential SearchStreamGang Example: Visualizing Hook Methods