Java Streams: Applying Spliterator

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 “Splittable iterators” (Spliterators)
- Recognize how to apply Spliterator to the SimpleSearchStream program

```
Search Words
"do", "re", "mi", "fa", "so", "la", "ti", "do"
```

```
stream()
map (this::searchForWord)
filter (not(SearchResults::isEmpty))
collect (toList())
```

See [github.com/douglascraigschmidt/LiveLessons/tree/master/SimpleSearchStream](https://github.com/douglascraigschmidt/LiveLessons/tree/master/SimpleSearchStream)
Applying Java Spliterator in SimpleSearchStream
Applying Java Spliterator in SimpleSearchStream

- The SimpleSearchStream program uses a sequential spliterator

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

```
stream()
map (this::searchForWord)
filter (not(SearchResults::isEmpty))
collect (toList())
```

See [github.com/douglascraigschmidt/LiveLessons/tree/master/SimpleSearchStream](https://github.com/douglascraigschmidt/LiveLessons/tree/master/SimpleSearchStream)
Applying Java Spliterator in SimpleSearchStream

- `searchForWord()` uses the spliterator to find all instances of a word in the input & return a list of all the SearchResults

```java
SearchResults searchForWord(String word) {
    return new SearchResults(..., word, ..., StreamSupport
        .stream(new WordMatchSpliterator(mInput, word),
        false)
        .collect(toList()));
}
```

See `SimpleSearchStream/src/main/java/search/WordSearcher.java`
searchForWord() uses the spliterator to find all instances of a word in the input & return a list of all the SearchResults

SearchResults searchForWord
    (String word)
    {
        return new SearchResults
            (... , word , ... , StreamSupport
                .stream(new WordMatchSpliterator
                    (mInput , word) ,
                    false)
                .collect(toList()));
    }

StreamSupport.stream() creates a sequential stream via the WordMatchSpliterator class

See docs.oracle.com/javase/8/docs/api/java/util/stream/StreamSupport.html#stream
Applying Java Spliterator in SimpleSearchStream

- `searchForWord()` uses the spliterator to find all instances of a word in the input & return a list of all the `SearchResults`

```
SearchResults searchForWord(String word){
    return new SearchResults(..., word, ..., StreamSupport.stream(new WordMatchSpliterator(mInput, word), false).collect(toList()));
}
```

- This stream is collected into a list of `SearchResults.Result` objects
Applying Java Spliterator in SimpleSearchStream

- WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string.

```java
class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {
    private final Matcher mWordMatcher;

    public WordMatchSpliterator(String input, String word) {
        String regexWord = "\b" + word.trim() + "\b";
        mWordMatcher = Pattern.compile(regexWord, Pattern.CASE_INSENSITIVE).matcher(input); ...
    }

    See SimpleSearchStream/src/main/java/search/WordMatchSpliterator.java
```
Applying Java Spliterator in SimpleSearchStream

- WordMatchSpliterator uses Java regex to create a stream of SearchResult objects that match the # of times a word appears in an input string

```java
class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {
    private final Matcher mWordMatcher;

    public WordMatchSpliterator(String input, String word) {
        String regexWord = "\\b" + word.trim() + "\\b";

        mWordMatcher = Pattern.compile(regexWord,
                                     Pattern.CASE_INSENSITIVE)
                        .matcher(input); ...
    }
```

The extending class need only implement `tryAdvance()`

See [docs.oracle.com/javase/8/docs/api/java/util/Spliterators.AbstractSpliterator.html](https://docs.oracle.com/javase/8/docs/api/java/util/Spliterators.AbstractSpliterator.html)
WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string.

class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {
    private final Matcher mWordMatcher;

    public WordMatchSpliterator(String input, String word) {
        ...
        String regexWord = "\\b" + word.trim() + "\\b";

        mWordMatcher = Pattern.compile(regexWord,
            Pattern.CASE_INSENSITIVE)
            .matcher(input); ...

"An engine that performs regex match operations on a character sequence."
Applying Java Spliterator in SimpleSearchStream

- WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string

```java
class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {
    private final Matcher mWordMatcher;

    public WordMatchSpliterator(String input, String word) {
        ...
        String regexWord = "\\b" + word.trim() + "\\b";

        mWordMatcher =
            Pattern.compile(regexWord,
                            Pattern.CASE_INSENSITIVE)
                .matcher(input); ...
    }
```

Constructor is passed the input string & a given word to search for matches.
Applying Java Spliterator in SimpleSearchStream

- WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string

```java
class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {
    private final Matcher mWordMatcher;

    public WordMatchSpliterator(String input, String word) {
        String regexWord = "\\b" + word.trim() + "\\b";
        mWordMatcher = Pattern.compile(regexWord, Pattern.CASE_INSENSITIVE).matcher(input); ...
    }
```

This regex only matches a "word"

See [www.vogella.com/tutorials/JavaRegularExpressions/article.html](http://www.vogella.com/tutorials/JavaRegularExpressions/article.html)
Applying Java Spliterator in SimpleSearchStream

- WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string.

class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<
private final Matcher mWordMatcher;

public WordMatchSpliterator(String input, String word) {
    ...
    String regexWord = "\\b" + word.trim() + "\\b";

    mWordMatcher =
        Pattern.compile(regexWord, Pattern.CASE_INSENSITIVE)
        .matcher(input); ...
Applying Java Spliterator in SimpleSearchStream

- WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string

```java
class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {

    ... 

    public boolean tryAdvance(Consumer<? super Result> action) {
        if (!mWordMatcher.find())
            return false;

        else {
            action.accept(new Result(mWordMatcher.start()));
            return true;
        }
    }
}
```

Called by the Java 8 streams framework to attempt to advance the spliterator by one word match

See docs.oracle.com/javase/8/docs/api/java/util/Spliterator.html#tryAdvance
Applying Java Spliterator in SimpleSearchStream

- WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string

```java
class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {

    ...

    public boolean tryAdvance(Consumer<? super Result> action) {
        if (!mWordMatcher.find()) {
            return false;
        } else {
            action.accept(new Result(mWordMatcher.start()));
            return true;
        }
    }
}
```

*Passes the result (if any) back "by reference" to the streams framework*
Applying Java Spliterator in SimpleSearchStream

WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string.

class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {
        ...
        public boolean tryAdvance(Consumer<? super Result> action) {
            if (!mWordMatcher.find())
                return false;
            else {
                action.accept(new Result(mWordMatcher.start()));
                return true;
            }
        }
    }

Check if any remaining phrases in the input match the regex
Applying Java Spliterator in SimpleSearchStream

WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string

class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {
        ...
        public boolean tryAdvance(Consumer<? super Result> action) {
            if (!mWordMatcher.find())
                return false;
            else {
                action.accept(new Result(mWordMatcher.start()));
                return true;
            }
        }
    }

Inform the streams framework to cease calling tryAdvance() if there's no match
Applying Java Spliterator in SimpleSearchStream

- WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string

```java
class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {

    ...

    public boolean tryAdvance(Consumer<? super Result> action) {
        if (!mWordMatcher.find())
            return false;
        else {
            action.accept(new Result(mWordMatcher.start()));
            return true;
        }
    }

    }
```
Applying Java Spliterator in SimpleSearchStream

- WordMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a word appears in an input string

```java
class WordMatchSpliterator
    extends Spliterators.AbstractSpliterator<Result> {
    ...

    public boolean tryAdvance(Consumer<? super Result> action) {
        if (!mWordMatcher.find())
            return false;
        else {
            action.accept(new Result(mWordMatcher.start()));
            return true;
        }
    }
}
```

Inform the streams framework to continue calling tryAdvance()
Applying Java Spliterator in SimpleSearchStream

Here’s a recap of how `searchForWord()` uses `WordMatchSpliterator`:

```java
SearchResults searchForWord(String word) {
    return new SearchResults(..., word, ..., StreamSupport
                            .stream(new WordMatchSpliterator
                                     (mInput, word),
                                     false)
                            .collect(toList()));
}
```

`StreamSupport.stream()` creates a sequential stream via the `WordMatchSpliterator` class.

Applying Java Spliterator in SimpleSearchStream

- Here's the output that searchForWord() & WordMatchMatchSpliterator produce

```
List<String>

Stream<String>

Stream<SearchResults>

Stream<SearchResults>

List<SearchResults>
```

- Search Words
  - "do", "re", "mi", "fa", "so", "la", "ti", "do"

- `stream()`
- `map (this::searchForWord)`
- `filter (not(SearchResults::isEmpty))`
- `collect (toList())`

45,000+ phrases

"do", "re", "mi", "fa", "so", "la", "ti", "do"
End of Java 8 Streams: Applying Spliterators