Java Streams: Short-Circuit Operations

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 stream aggregate operations
  - Intermediate operations
  - Terminal operations
  - Short-circuit operations

\[ \xi \rightarrow \text{Input } x \]

\( \text{takeWhile (predicate)} \)

\( \text{Output } f(x) \)

\( \text{limit(maxSize)} \)

\( \text{Output } g(f(x)) \)

\( \text{findAny()} \)
Java 8 Streams Short-Circuit Operations
Java 8 Streams Short-Circuit Operations

• An aggregate operation *may* process all elements in a stream
Java 8 Streams Short-Circuit Operations

- An aggregate operation *may* process all elements in a stream, e.g.
  - map() processes all of the elements in its input stream

```
Optional findFirst()
Stream limit(long maxSize)
Stream takeWhile(Predicate<...> p)
Stream map(Function<...> mapper)
```

Input x
Output f(x)
Output g(f(x))
Output g(f(x))
Java 8 Streams Short-Circuit Operations

- An aggregate operation *may* process all elements in a stream, e.g.
- map() processes all of the elements in its input stream
- Unless a behavior throws an exception..

See [vanilla-java.github.io/2016/06/21/Reviewing-Exception-Handling.html](vanilla-java.github.io/2016/06/21/Reviewing-Exception-Handling.html)
Java 8 Streams Short-Circuit Operations

- An aggregate operation *may* process all elements in a stream, e.g.
  - `map()` processes all of the elements in its input stream
  - “Short-circuit” operations halt further processing after reaching their condition

Java 8 Streams Short-Circuit Operations

- An aggregate operation *may* process all elements in a stream, e.g.
  - map() processes all of the elements in its input stream

- "Short-circuit" operations halt further processing after reaching their condition
  - takeWhile()
    - A short-circuit intermediate operation that returns a stream consisting of a subset of elements taken from this stream that match the given predicate

See [docs.oracle.com/javase/9/docs/api/java/util/stream/Stream.html#takeWhile](http://docs.oracle.com/javase/9/docs/api/java/util/stream/Stream.html#takeWhile)
An aggregate operation *may* process all elements in a stream, e.g.

- map() processes all of the elements in its input stream
- “Short-circuit” operations halt further processing after reaching their condition
  - takeWhile()
  - limit()
    - A short-circuit intermediate operation that causes a stream to operate on a reduced size

See [docs.oracle.com/javase/8/docs/api/java/util/stream/Stream.html#limit](docs.oracle.com/javase/8/docs/api/java/util/stream/Stream.html#limit)
An aggregate operation *may* process all elements in a stream, e.g.

- `map()` processes all of the elements in its input stream
- "Short-circuit" operations halt further processing after reaching their condition
  - `takeWhile()`
  - `limit()`
  - `findFirst()`, `findAny()`, `anyMatch()`, `allMatch()`, & `noneMatch()`
- Short-circuit terminal operations can finish before traversing all elements in the underlying stream

See [dzone.com/articles/collectors-part-1-%E2%80%93-reductions](http://dzone.com/articles/collectors-part-1-%E2%80%93-reductions)
End of Java Streams: Short-Circuit Operations